measurement systems laboratory

massachusetts institute of technology, cambridge. massachusetts 02139

N71-34414

TE-46

PARAMETER UNCERTAINTIES IN CONTROL SYSTEM DESIGN

by

Thorgeir Palsson

May 1971

TE-46

PARAMETER UNCERTAINTIES IN CONTROL SYSTEM DESIGN

by

Thorgeir Palsson

May 1971

Measurement Systems Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts

Approved: W. Markey

Director
Measurement Systems Laboratory

TE-46

PARAMETER UNCERTAINTIES IN CONTROL SYSTEM DESIGN

by

Thorgeir Palsson

S.B.(1966), S.M.(1967)

Massachusetts Institute of Technology

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF SCIENCE

at the

Massachusetts Institute of Technology

May 7, 1971

Signature of Author: Thorgeir Palsson
Department of Aeronautics and Astronautics
May 1971

Certified by: H. Philip Whitaker
Thesis Supervisor

Certified by: John L. Deyst Jr.
Thesis Supervisor

Certified by: Walter Wrigley
Thesis Supervisor

Certified by: James E. Potter
Thesis Supervisor

Accepted by: Judson J. Baron
Chairman, Departmental Graduate Committee

PARAMETER UNCERTAINTIES IN CONTROL

SYSTEM DESIGN

by

Thorgeir Palsson

Submitted to the Department of Aeronautics and Astronautics in May 1971 in partial fulfillment of the requirements for the degree of Doctor of Science.

ABSTRACT

A design method is developed for including the effects of parameter uncertainties in the design of linear control systems. The approach taken to this problem may be classified as a special case of the stochastic control problem. Thus, the formulation is based on the minimization of the expected value of a quadratic performance index defined in terms of the system state vector. The uncertainty in the value of the performance index is then the result of the statistical nature of the system parameters rather than a random input signal. It is shown that the expected value of the performance index may be written as a sum of two terms under the assumption of first order variations of the system state. The first of these terms expresses the nominal performance of the system when the system parameters assume their mean values. The second term represents the effect of the uncertainties on the expected value of the performance index, and is interpreted as an index of system sensitivity. The total performance index is minimized with respect to designated free design parameters in a fixed configuration. The key to the numerical solution of this problem lies in using the phase-variable form of the system equations. Very efficient numerical techniques are developed for obtaining this solution using a gradient algorithm. The method is finally applied, with considerable success, to the design of two flight control systems.

Thesis Supervisors:

H. Philip Whitaker
Title: Professor of Aeronautics and Astronautics

Walter Wrigley, Sc.D.
Title: Professor of Instrumentation and Astronautics

James E. Potter, Ph.D.
Title: Associate Professor of Aeronautics and Astronautics

John J. Deyst, Sc.D.
Title: Associate Professor of Aeronautics and Astronautics

ACKNOWLEDGMENTS

I would like to express my appreciation to my thesis committee, Professor Philip Whitaker, Professor Walter Wrigley, Professor James Potter and Professor John Deyst, for their advice and constructive criticism during this thesis research. My sincerest thanks go to Professor Whitaker who has given freely of his time during this effort in numerous fruitful discussions and by reading the first draft of this document. Special thanks are also extended to Professor Wrigley for his advice and counsel throughout my graduate study and to Professor Potter whose suggestions on the numerical techniques have been invaluable.

I would also like to express my thanks to these gentlemen: Professor Winston Markey who as the director of the Measurement Systems Laboratory has provided a stimulating environment for this research; Dr. Kenneth Britting under whose contract this effort was initiated; Dr. Herman Rediess of the NASA Flight Research Center, for his interest in this effort and for providing the support for its completion, and Professor Walter Hollister for his interest in my graduate study. I am also indebted to the board members of the Icelandic Science Foundation who provided part of my personal support.

Special recognition is also due Mrs. Jennifer Kelley for her expert typing of this document and Mrs. Ann Preston for supervising and preparing its publication.

Lastly, I would like to thank my family for their patience during this effort and especially my wife, Anna, for her encouragement. The understanding and support of my parents throughout my education is deeply appreciated.

This thesis was prepared under D.S.R. Project 70343 and D.S.R. Project 72709 sponsored, respectively, by the National Aeronautics and

Space Administration, Electronics Research Center, Cambridge, Massachusetts, and Flight Research Center, Edwards, California, through NASA Grants No. NGR 22-009-229 and No. NGR 22-000-548.

The computational work of this thesis was done at the Information Processing Center at M.I.T.

TABLE OF CONTENTS

LIST OF SYMBOLS

Symbol	Definition
$\underline{A}$	system coefficient matrix
$\underline{E}_i$	$i^{\underline{th}}$ order matrix polynomial
G	system transfer function
J	system performance index
J_s	sensitivity performance index
$\underline{P}$	matrix of Lagrange multipliers
P	plant transfer function
$\underline{Q}$	state weighting matrix
$\underline{R}$	parameter covariance matrix
$\underline{S}$	sensitivity weighting matrix
S_{OL}	open-loop static sensitivity
$\underline{V}$,$\underline{W}$	system coefficient covariance matrices
$\underline{X}$,$\underline{Y}$,$\underline{Z}$	matrices of system state integrals
$\underline{a}$	system characteristic coefficient vector
a_i	$(i-1)^{\underline{st}}$ element of $\underline{a}$
$\underline{b}$	numerator coefficient vector of system transfer function
b_i	$(i-1)^{\underline{st}}$ element of $\underline{b}$

Symbol	Definition
$\underline{c}$	system input vector
$\underline{g}$	gradient vector
k	number of model zeros
l	order of model equation
m	number of system zeros
n	order of system equation
$\underline{p}$	vector of free design parameters
p_i	$i^{\underline{th}}$ element of $\underline{p}$ or the $i^{\underline{th}}$ system pole
u	scalar system input
$\underline{x}$	system transient state vector
$\delta\underline{x}$	variation of transient state vector
$\underline{y}$	system output state vector
$\delta\underline{y}$	variation of output state vector
y	scalar output of system
z_i	$i^{\underline{th}}$ system zero
$\underline{\Lambda}_i$	matrix of Lagrange variables
Δ	represents the change of error of a quantity
$\underline{\alpha}$	model characteristic coefficient vector
α_i	$(i-1)^{\underline{st}}$ element of $\underline{\alpha}$

Symbol	Definition
$\underline{\tilde{\alpha}}$	extended model coefficient vector
$\underline{\beta}$	numerator coefficient vector of model transfer function
β_i	$(i-1)^{\underline{st}}$ element of $\underline{\beta}$
δ	indicates the variation of a quantity due to variable design parameters
$\tilde{\delta}$	indicates the variation of a quantity due to free design parameters
ε	adjustable weighting factor
$\underline{\eta}$	n-dimensional vector all of whose elements are zero except the last one, which is equal to unity
μ	trade-off parameter
$\underline{\xi}$	vector of variable design parameters
$\underline{\sigma}$	vector of sensitivity functions

SPECIAL NOTATION

$(\dot{\ })$	indicates derivative with respect to time
$(\ )^{(i)}$	$i^{\underline{th}}$ derivative of quantity
$(\hat{\ })$	indicates that quantity is associated with model
$(\tilde{\ })$	indicates extended dimension of a vector or matrix
$(\ )^T$	transpose of a matrix
*	indicates that quantity is evaluate for the nominal parameter values

Symbol	Definition
0	subscript denoting initial value of quantity
ss	subscript indicating steady-state value

Chapter 1. Introduction and Summary

1.1 Introduction

It is well known that one of the principal reasons for using feedback in control systems is to reduce the effects of undesirable disturbances of various origins on the system performance. The most common disturbances of this type are unwanted inputs and variations of the static and dynamic characteristics of the system.

It is important that the disturbance effects be accounted for in the design process in addition to achieving desirable response characteristic to input commands. A number of standard methods are available for controlling the effects of unwanted input signals, which may be deterministic or random in nature. Considerable effort has also been made over the past two decades to develop design techniques, which make it possible to achieve satisfactory system performance despite changes or uncertainties in the dynamic characteristics of the control system or its mathematical description.

This effort has proceeded in two different directions, i.e. towards adaptive systems on the one hand and insensitive systems on the other.

In the case of adaptive control systems adjustment can be made in the controller in order to cope with changes in the dynamics of the controlled member of the system. These adjustments can be made by detection of changes in the system response or as a function of changes in some environmental parameters, which affect the system in a known way. In contrast, the insensitive system should be capable of achieving satisfactory performance for all anticipated operating conditions without any adjustments of the controller characteristics.

Deciding which of these two types of systems should be used in a given application may not always be a simple one, since the differences in the capabilities of the two types have not been clearly identified.[40] It is a reasonable assumption, however, that a control system will always be required to exhibit a certain degree of insensitivity to small variations of its dynamics, without the need for any adjustments of its parameters.

The subject of system sensitivity focuses on the description and analysis of the effects of variations and inaccuracies in the system characteristics on its performance. As pointed out in Reference [42] this is a problem which is peculiar to engineering design, because of the differences which exist between the mathematical model and the actual system. Thus, the designer must concern himself with the influence of inaccurately known system parameters on the design. Such parameters commonly change their values over a period of time, although they may be assumed to be invariant in the system design. Manufacturing tolerances and changes in the operating environment are also sources of uncertainty which must be considered in any control system design.

A number of very useful definitions and techniques have been developed for the analysis of these effects in linear systems. The now classical definition of system sensitivity was given in Reference [1] as the change in the closed-loop transfer function with respect to changes in the transfer function of the plant. This definition of system sensitivity can be used to compute the changes in the system frequency response due to specified changes in the plant. It also provides a useful relationship between the sensitivity of the frequency response and the loop gain at any given frequency. An extension of this definition of multivariable systems is given in Reference [6]. Design methods for achieving a frequency response which satisfies stated tolerances have been developed in References [15], [16] and [25]. The advantage of these frequency domain methods is that their application is not restricted to small parameter variations. However, they do require that all the system specifications be stated in terms of the tolerances on the frequency response or be transformable into that form. Their usefulness, when the transient time response is of primary interest, is therefore open to question. Difficulties are also encountered in the case of non-minimum phase plants, when the relationship between the amplitude and phase of the frequency response is no longer unique. References [19] and [25] demonstrate the use of these methods in the design of flight control systems.

The emphasis on root locus techniques in control system design led to the definition of the closed-loop pole sensitivities as the derivatives of these poles with respect to the open-loop gain, poles and zeros.[26][46] By computing these sensitivities for any given closed-loop pole it is possible to determine the incremental change in its location due to variations in the open-loop parameters. These sensitivities and the associated techniques for their computation are powerful tools for analyzing the effect of system changes on the closed-loop roots. As such they can also be used in the design process to predict the effects of varying any free design parameters on the closed-loop behaviour. Techniques for achieving favourable closed-loop sensitivities have been developed in References [16], [17], and [20]. Their application is most useful in the case of systems with relatively few dominant modes whose location in the complex plane can be specified in terms of bounded areas. By strategically locating the singularities of the compensating components, the movement of the dominant system roots due to open-loop changes can be restricted.

The common characteristics of the design methods in the domain of real and complex frequencies are that high feedback gain is used in the appropriate frequency band to suppress the effect of changes which occur in the plant, i.e. in the forward path. It is then implicitly assumed that the properties of the compensation are highly stable compared with the plant, since the high feedback gain has the effect of amplifying any changes occurring in that path. The problems of system stability and noise are considered as constraints, which determine the character of the compensation at high frequency. The sensitivity problem is, furthermore, separated from the achievement of an acceptable nominal response by constraining the response in the important frequency region. Although these techniques have been demonstrated to satisfy requirements of the type mentioned above, one suspects that the method of separating the sensitivity problem from the remaining system specifications may result in unnecessarily complicated systems. An excellent review of these techniques is available in Reference [12].

The increasing importance of state-space and optimal control techniques has resulted in numerous papers on system sensitivity in the time domain. Much of this research is based on the use of the sensitivity functions, which are defined as the derivatives of the system state variables with respect to the variable system parameter under consideration.[41] The sensitivity functions are therefore measures of the deviation of the system response from its nominal due to variations of the corresponding parameter. A number of papers [2],[4],[7],[8],[10],[21],[22],[23],[36] have appeared in recent years on the application of a quadratic sensitivity index, defined in terms of the sensitivity functions, to the optimal design of systems which are subject to variations of some plant parameters. This sensitivity index is added to the quadratic performance index, which has been chosen to optimize the system's nominal response. By minimizing this sum with respect to the available control inputs it is hoped that the resulting system design exhibits sensitivity properties, which are more favourable than would have been the case if the sensitivity index had not been included. The difficulty arises in solving for the optimum control inputs because of the desire to determine these inputs in the form of feedbacks of the system state variables. The variations of these state variables from their nominal responses then result in corresponding variations of the feedback control signals. If the form of these signals remains to be determined, however, the effects of the control variations on the system sensitivity functions cannot be determined.

For this reason, it is necessary to specify the form of the feedback control, which is usually taken to be a linear feedback of all the system states and their sensitivity functions with the values of the gains free to be chosen. Some further assumptions must also be made about the second order effects of the parameter variations on the system response.[37] Because of the need to feedback the sensitivity functions in addition to the system states, the resulting system design becomes very complex even for a simple system and has in some cases led to inconclusive results.[36] This complexity is avoided by formulating the control as a function of the state variables only, which has been applied with some success to the design of an attitude control system of a booster.[15][33]

The basic problem with using the sensitivity functions for the definition of an index of sensitivity is that a separate set of these functions must be defined for each variable parameter, which is to be considered. This means that it is very difficult to obtain the numerical solution for multiple parameter variations. As a result, only single parameter variations have been considered in the application of the methods just described. The choice of the weighting matrix in the sensitivity index is also open to question, since no systematic method has been proposed for its selection. Very little effort has been made to relate the results of using the sensitivity index to realistic specifications on the time response of the system.

A somewhat different approach to the problem of system sensitivity was proposed in Reference [24], which formulated a sensitivity index in terms of the mean square deviation of the system response to changes in a system parameter. The method suffers from the numerical difficulties encountered in determining the type of compensation networks that are required to minimize the sensitivity index. A more general approach is taken in Reference [43], which formulates the problem as an optimal stochastic control problem with changes in the system parameters described as random variations. However, the resulting optimal control is open-loop and cannot, in general, be put into a feedback form. Parameter identification techniques have also been applied recently to the problem of uncertain parameters in non-linear control systems [38].

Finally, the sensitivity problem has been analyzed in terms of the effect of parameter variations on the value of the performance index [9],[28],[29]. Sensitivity indices based on the variation of the performance index have been defined in References [3] and [45] with application to simple control systems. Both of these studies are preoccupied with the value of the performance index and its deviation due to the parameter variations, but fail to interpret the impact on the system time response or how their techniques may be applied to realistic design problems.

The objective of this thesis is to develop a sensitivity design method in the time domain, which alleviates some of the difficulties, which have been encountered in the studies cited above. In particular, it is clear that if the solution of practical design problems is to be attempted it is necessary to develop computational techniques, which can be applied to high order systems efficiently and in a straightforward manner. A systematic method for choosing the measure of sensitivity is also important for the same reasons.

Furthermore, it should be possible to include any number of variable design parameters and these should not be restricted to parameters of the system plant. It is also desirable that the complexity of the system be left to the designer as opposed to a priori specifications of all feedback loops.

1.2 Problem Approach

The approach which is taken here to the problem of sensitivity in control system design may be classified as a special case of stochastic control system design. Thus, it is based on the notion that the effect of random disturbances on the system performance may be accounted for by defining the index of performance as the expected value of the functional which describes the system performance in the absence of these disturbances. The assumption is made that the system is described by linear differential equations with coefficients whose values may be inaccurately known or are subject to changes, which are slow relative to the response time of the system. These coefficients which will be referred to as design parameters, may therefore be considered to be time invariant and statistically distributed about some nominal values. In addition, the following assumptions are made:

(1) the variations of the design parameters have a joint probability distribution with known first and second order statistics

(2) the parameter variations are small enough so that the corresponding deviations of the system response may be described by a first order approximation

(3) the performance index is a quadratic integral in terms of the system state vector

(4) the configuration of the system is specified a priori with designated free design parameters, which may be optimized

(5) the system specifications may be stated in terms of a desirable time response to a step input

Assumption (2) is necessary in order to make the computational task tractable and has been used in most of the studies of time domain sensitivity, which have been reviewed in the course of this work. Its effect is to allow a certain degree of separation of the equations describing the nominal system response from the equations describing its deviations. The use of the quadratic performance index is justified on the basis of its widespread acceptance in control system design. A new dimension has also been added to its usefulness in satisfying specific response requirements by Reference [31], which provides a systematic technique for selecting the state weighting matrix. The integral square error criterion has also been found to be an effective tool for this purpose.[47]

Assumption (4) is made in the interest of avoiding the problems involved in determining the effect of parameter variations on a yet to be determined control input, in addition to allowing the designer to limit the complexity of the system. The formulation of the fixed configuration technique follows closely the development given in Reference [31], which is based on the use of the state equations of the transient response of the closed-loop system. The assumption that the system response specifications be stated in terms of a model step response is a matter of convenience for determining the value of the state weighting matrix but does not restrict the application of the method to any particular choice of this weighting matrix.

The design method is developed for a single input/output linear control system, whose closed-loop characteristics are represented by a transfer function of the form:

$$\frac{y(s)}{u(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1} s + \ldots + a_1 s + a_0} \quad m<n \qquad (1.1)$$

The coefficients of the numerator and denominator therefore become functions of the system design parameters:

$$\underline{a} = \underline{a}\ (\ \underline{p},\ \underline{\xi}\ )$$

$$\underline{b} = \underline{b}\ (\ \underline{p},\ \underline{\xi}\ )$$

where $\underline{a}$ and $\underline{b}$ contain the coefficients of Equation (1.1), and $\underline{p}$ and $\underline{\xi}$ are the vector representations of the free and variable design parameters respectively. Actually, $\underline{p}$ and $\underline{\xi}$ may contain common elements, which means that free design parameters with uncertainties or variations about a nominal value can be considered. A simple but convenient method for handling this case is developed in the body of the thesis whereby the free design parameter is represented as a product of its nominal value and a random parameter with the mean value of unity.

The transfer function of Equation (1.1) can be transformed into first order state equations in a number of ways. A particularly convenient form which is used in this work is given by:

$$\dot{\underline{y}} = \underline{A}\ \underline{y} + \underline{c}\ u \quad ; \quad \underline{y}(0) = \underline{0} \qquad (1.2)$$

where the first element of $\underline{y}$ is identical to the system output. The system matrix is in the phase-variable form, i.e. contains only ones and zeros except for the last row which consists of the denominator coefficients in Equation (1.1). The input vector, $\underline{c}$, is a function of both the numerator and denominator coefficients. This is sometimes referred to as the standard observable realization of the transfer function. The transient response of he system may be obtained from Equation (1.2), assuming a unit step input, in the form:

$$\dot{\underline{x}} = \underline{A}\,\underline{x} \quad ; \quad \underline{x}(0) = \underline{x}_0 \tag{1.3}$$

where $\underline{x}$ is the difference between $\underline{y}$ and its steady-state value and $\underline{x}_0$ is a function of the transfer function coefficients. This is the form used in Reference [32] to represent the system response, although its development differs somewhat from the one given in this thesis.

1.3 Synthesis of Results

The problem of parameter uncertainty or variation has been formulated as the minimization of the expected value of a quadratic performance index of the form:

$$\bar{J} = \overline{\int_0^\infty \underline{x}^T\,\underline{Q}\,\underline{x}\,dt} \tag{1.4}$$

where $\underline{Q}$ is a constant positive semi-definite weighting matrix. By using the assumption about the linearity of the deviations of the response due to parameter variations, it was shown that this performance index can be written as the sum of two terms:

$$\bar{J} = \text{trace}\left[\underline{Q}\left[\int_0^\infty \underline{x}_*\,\underline{x}_*^T\,dt + \int_0^\infty \overline{\delta\underline{x}\,\delta\underline{x}^T}\,dt\right]\right] \tag{1.5}$$

where $\underline{x}_*$ represents the nominal system response based on the mean values of the variable design parameters and $\delta\underline{x}$ denotes the first order deviation of the response due to a variation of the variable design parameters. The first of these terms expresses the nominal performance of the system. The second term represents the effect of the parameter variations on the performance and can be interpreted as an index of system sensitivity. Thus, this formulation relates the stochastic approach to the technique of adding a sensitivity index to the performance index. The form of the sensitivity index is also completely specified as a function of the total deviation of the system state due to the simultaneous variation of all the variable design parameters.

The problem of computing the numerical value of $\overline{J}$, given the first and second order statistics of the variable design parameters, is very difficult in general. This is so mainly because it is necessary to obtain the integral of the covariance matrix $\overline{\delta\underline{x}\ \delta\underline{x}^T}$ over all time as seen from Equation (1.5). In most cases, this would require the numerical integration of the matrix differential equation describing the propagation of this matrix in time. The phase-variable form of the state equations, however, allows this integral to be obtained as the solution of n+2 linear algebraic matrix equations where n is the order of the system. Specifically, these equations may be written :

$$\underline{A}_* \underline{X} + \underline{X}\,\underline{A}_*^T + \underline{X}_0 = \underline{0}$$

$$\underline{A}_* \overline{\delta\underline{X}} + \overline{\delta\underline{X}}\,\underline{A}_*^T + \underline{U}\ (\underline{Z}_i) + \overline{\delta\underline{X}_0} = \underline{0}$$

$$\underline{Z}_i + \underline{A}_* \underline{Z}_{i-1} + \underline{x}_0^* \underline{v}_{i-1}^T = \underline{0} \qquad (1.6)$$

$$\underline{A}_* \underline{Z}_{n-1} - a_{n-1}^* \underline{Z}_{n-1} - \ldots - a_0^* \underline{Z}_0 + \underline{x}_0^* \underline{v}_{n-1}^T - \underline{X}\,\underline{W} = \underline{0}$$

where $\underline{X}$ and $\overline{\delta \underline{X}}$ denote the two integral terms in Equation (1.5) and $\underline{U}$ is a function of the unknown $\underline{Z}_i$ matrices as indicated. The remaining terms are all functions of the system coefficients and the covariance matrix of the variable design parameters. These terms are evaluated for the nominal values of these parameters as indicated by the asterisk.

The first two of these equations form a special case of the well known Riccatti equation. The n remaining matrix equations express the effect of the correlation between the various terms of $\delta \underline{x}$ and $\delta \underline{a}$ on the solution of the sensitivity index, where $\delta \underline{a}$ represents the variation of the system's characteristic coefficient vector. It is not clear that these equations can be solved in any convenient way. A numerical integration technique has typically been used to obtain the solution of the steady-state Riccatti equation.[31] This would be a prohibitive computational task in this case for any practical system in view of the number of equations and their interdependence.

The key to the simplification of the numerical solution of Equations (1.6) lies in the phase-variable form of the system matrix $\underline{A}$. By writing the first two equations column by column an iterative relationship between (n-1) of the column vectors of the solution matrices is obtained. The form of these column vector equations is similar to that of the equations for $\underline{Z}_i$ in Equation (1.6), which was also obtained by taking advantage of the form of the system matrix. By successive substitution of these iterative expressions, explicit expressions are obtained for the unknown matrices in Equation (1.6). A single matrix inversion of an (n×n) matrix is required to obtain all these solutions.

Another n+2 matrix equations are added to the set of equations which must be solved in order to determine the gradient of

$\bar{J}$ for any given set of the free design parameters. Their solutions are obtained in a similar way to those of Equations (1.6). A steepest descent algorithm is then used to determine the local minimum of $\bar{J}$. This means that 2(n+2) matrix equations containing (n×n) matrices must be solved for each iteration of the algorithm. Highly efficient computer programs have been developed for this purpose. In most practical problems the first solution of Equation (1.6) is not sufficiently accurate due to round-off errors. A very successful iterative process has been used to refine the solutions to an accuracy of better than one part per 10^{10} using double precision.

The design method described above has been applied to design examples using the model performance index developed by Rediess.[31] The advantage of this index over other quadratic indices is that its weighting matrix can be determined in a systematic way, once the transfer function of the desired model response has been chosen. A new interpretation of this performance index is given in this thesis, where it is shown that the model performance index may be written as:

$$\mathrm{MPI} = \int_0^\infty i^2(t)\, dt \tag{1.7}$$

where i(t) is the scalar input to an error model describing the difference in the responses of the system and the reference model which describes the desired system response. In its original derivation it is necessary, in general, to add a second term to the model performance index when the system transfer function contains zeros. This term is defined as a weighted quadratic form of the error in the initial state as compared with the model's initial state. Some ambiguity is caused by the arbitrary weighting of this term relative to the integral of Equation (1.7). A technique which eliminates the need for this second term of the performance index has been developed using the new interpretation. This is accomplished by an expansion of the system order and by requiring the transfer function of the reference model to have a number of excess poles over zeros which is equal to or smaller than that of the system.

The basic approach which is taken in applying the present design method may be stated in terms of the following steps:

(1) the configuration is chosen with the objective of obtaining a satisfactory nominal response

(2) the free design parameters are optimized by determining the minimum value of the nominal performance index, i.e. assuming that all the design parameters are known and invariant

(3) the expected value of the performance index is minimized for a specified value of the parameter covariance matrix and the solution compared with the solution of step number (2)

In most cases step number (3) will decrease the value of the sensitivity index when compared with the sensitivity index achieved by step number (2). The amount of reduction can be controlled to some extent by varying the effect of the uncertainty on the performance index. This is done most conveniently by scaling the covariance matrix of the variable parameters, i.e. increasing or decreasing the amount of uncertainty in the values of these parameters. The relationship between the individual parameter variations is unaffected by the scaling.

In most practical problems the improvement in the sensitivity index by the third design step is achieved by some deterioration of the nominal performance index from that achieved by step number (2). It is desirable that this deterioration be small relative to the change in the sensitivity index in order to achieve an overall improvement in the system response, which is expressed by the total change in expected value of the performance index. In some problems, however, it may be of more importance to reduce the sensitivity of the system than achieving the desired response characteristics. This is clearly the case when the variation of a design parameter may result in an unstable response. If the design which is obtained by step number (3) is still too sensitive, i.e. has too large deviations for the specified parameter variations, then a new configuration must be chosen and the process repeated. It is emphasized that the performance indices are only tools which may be used to achieve the desired results. Thus, the system design must always be judged on the basis of comparison of its response with the original specifications. The relative changes in the performance indices can, however, be used to estimate the corresponding changes in the actual system performance.

The sensitivity design method presented here has been applied to the design of two flight control systems. The first of these is an attitude control system of a flexible booster whose bending characteristics are inaccurately known. The nominal design, which was obtained without any regard for the uncertainties, was found to be very sensitive to changes in the bending frequency and to a lesser extent in the bending mode slope. This sensitivity was significantly reduced by application of the described technique resulting in a stable response for a much larger range of parameter variations than achieved by the nominal design. The second system is an attitude control system of a high performance aircraft with two structural bending modes included in the vehicle dynamics. The frequencies of the bending modes and the dimensional coefficients of the rigid body dynamics are all described by normal distributions about their mean values. The nominal design of this system was found to provide insufficient damping of the first bending mode. By taking advantage of the system's sensitivity to changes in the bending frequency it was found possible to significantly reduce the effect of this mode on the response. No improvement could be obtained in the system's sensitivity to changes in the dimensional coefficients. The nominal design was found to be inherently insensitive to these changes as indicated by a small value of the sensitivity index relative to the nominal performance index.

1.4 Conclusions

The following conclusions have been reached in this research:

(1) It has been shown that the sensitivity design method, developed on the basis of minimizing the expected value of a quadratic performance index, can be used to design control systems which are less sensitive to uncertainties in the system parameters than the designs obtained by minimizing the corresponding deterministic performance index, using the nominal values of the system parameters.

(2) The expected value of the quadratic performance index may be written as the sum of two terms. The first of these represents the nominal system performance and the second term may be interpreted as an index of system sensitivity.

(3) The computational difficulties which have been associated with previous sensitivity design methods have been alleviated allowing any number of correlated parameter variations to be handled with ease.

(4) The improvement in the overall performance of the system can be rated by a parameter, μ, expressing the trade-off between the change in sensitivity and the change in nominal performance. A low value of μ indicates that the reduction in sensitivity is achieved at a high cost in terms of changes in the nominal system performance. For a large μ the opposite holds true.

(5) By computing the sensitivity index corresponding to individual system parameters the relative effect of uncertainties in these parameters on the system performance can be measured. The values of the sensitivity indices can be used to estimate the allowable tolerances of these parameters.

1.5 Recommendations for Further Research

It is felt that the sensitivity method presented in this thesis may be applicable to a much wider range of problems than have been considered here. A broad investigation of its potential in control system design would, therefore, be of interest. In the case of the model performance index, it would be desirable to investigate further the effect of the model on the resulting solution. It has been found, for instance, that the sensitivity of the nominal solution may vary considerably depending on what type of model is used. The application of the sensitivity method to other quadratic performance indices is also an open area for further study.

Finally, the computer programs which have been written in the course of this work could be developed into a versatile design package based on the minimization of any desired quadratic performance index with respect to the free design parameters of a linear, fixed configuration control system. A relatively moderate effort would be required for this purpose.

Chapter 2. Control System Sensitivity

2.1 Introduction

The area of system sensitivity, which will be addressed here, is concerned with the effects of changes in the static and dynamic characteristics of the system on its response to specified inputs. These changes can often be described in terms of variations of some system design parameters, which may represent actual changes in the system characteristics or inaccuracies in their knowledge. These variations are typically assumed to be time invariant over the time interval of interest, with a specified probability distribution.

Some fundamental definitions of system sensitivity have been developed in the literature in order to systematically analyze the effects of variations of this type. It is plausible that the definition of sensitivity which is most useful in each situation depends to a great extent on the form in which the system performance specifications are expressed. Accordingly the three basic definitions of system sensitivity are given in the time domain and in the frequency domain in terms of real and complex frequencies. These are, respectively:

- the sensitivity functions, which are the derivatives of the system state variables with respect to the variable design parameter
- the transfer function sensitivity, expressing differential changes in the system transfer function due to variations of some design parameters
- the sensitivity of closed-loop poles to changes in the open-loop static sensitivity, poles and zeros

As the objective of this thesis is to develop a practical method for including the problem of system sensitivity in the design process, it is appropriate to consider these definitions of system sensitivity in some detail. Their application to control system design will be examined for a single input/output system of the type shown in Figure 2.1.

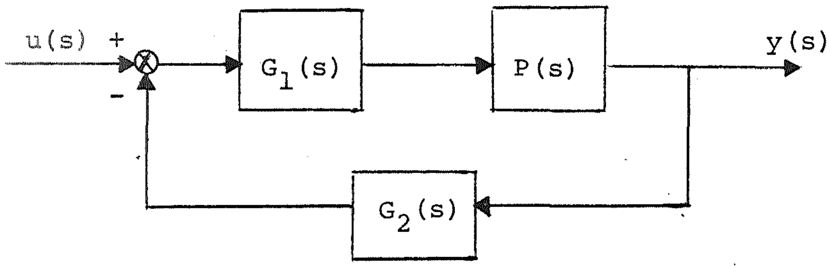


Figure 2.1. Single input/output system.

The plant, which is to be controlled, is represented by P(s); $G_1(s)$ and $G_2(s)$ are the transfer functions of the controller components. It is assumed that the parameter changes may occur in any of these elements.

Not surprisingly, the three types of system sensitivity are interrelated although the relationship is not necessarily simple.

2.2 Time Domain Sensitivity

System specifications in the time domain have been emphasized in recent years by the increasing use of state-space and optimal control techniques in the design of control systems. Specifications of this type are basic in the sense that it is by observation of the time behaviour of the system variables that one judges the ability of the control system to perform its function. Thus, the common characteristic of these specifications is that they describe the desirable behaviour of the system state variables with varying degrees of complexity. For instance, the system response may only be required to be stable, or it may be completely prescribed as a function of time.

The sensitivity function has been found to be a useful tool for determining the effect of small parameter variations, or uncertainties, on the system time response. This function is, in general, defined in terms of the system state vector. The corresponding state equation may be obtained from the transfer function of a single input/output system which is written in the following form:

$$\frac{\underline{y}(s)}{u(s)} = \frac{b_m s^m + \ldots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} \tag{2.1}$$

One possible state-space realization of this transfer function is given by:

$$\dot{\underline{y}}(t) = \underline{A}\,\underline{y}(t) + \underline{c}\,u(t) \quad ; \qquad \underline{y}(0) = \underline{0} \tag{2.2}$$

where $\underline{y}$ is an n-dimensional state vector, and $\underline{A}$ is an n×n system matrix in the phase variable form:

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 \ldots & 0 \\ 0 & 0 & 1 \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 \ldots & 1 \\ -a_0 & -a_1 & \ldots & -a_{n-1} \end{bmatrix}$$

and the elements of $\underline{c}$ are given by the following equation:

$$c_i = 0 \qquad 1 \le i < n-m$$

$$c_i = b_{n-i} - \sum_{j=n-m}^{i-1} a_{j+n-i} c_j \qquad n-m \le i \le n \tag{2.3}$$

where it is assumed that $m < n$, i.e. the system transfer function has at least one more pole than zeros. This is sometimes referred to as the standard observable realization of the transfer function given by Equation (2.1).

The sensitivity function of the state vector with respect to a single variable parameter, is an n-dimensional vector function defined by:

$$\underline{\sigma}(t) = \frac{d\underline{y}(t)}{d\xi/\xi} \tag{2.4}$$

where ξ denotes the variable parameter. The differential equation for $\underline{\sigma}(t)$ is obtained by differentiating Equation (2.2) with respect to ξ

$$\dot{\underline{\sigma}}(t) = \underline{A}\,\underline{\sigma}(t) + \frac{\partial\,\underline{A}}{\partial\xi/\xi}\,\underline{y}(t) + \frac{\partial\underline{c}}{\partial\xi/\xi}\,u(t) \tag{2.5}$$

where it is assumed that $u(t)$ is an external input and is, therefore, not affected by the parameter change. If $u(t)$, on the other hand, is a function of the system state, as in the case of a closed-loop feedback, this equation is:

$$\dot{\underline{\sigma}}(t) = \underline{A}\,\underline{\sigma}(t) + \frac{\partial\,\underline{A}}{\partial\xi/\xi}\,\underline{y}(t) + \frac{\partial\,\underline{c}}{\partial\xi/\xi}\,u(\underline{y},t) + \underline{c}\,\frac{\partial u(\underline{y},t)}{\partial\xi/\xi} \tag{2.6}$$

In order to compute $\underline{\sigma}(t)$ for this closed-loop system, the relationship between $u(t)$ and $\underline{y}(t)$ must, therefore, be known.

This poses a dilemma, when the objective is to determine the control input that optimizes a performance index containing $\underline{\sigma}(t)$. The difficulty is usually avoided by specifying the functional relationship of u(t) and the system variables.

2.2.1 Sensitivity Index

The use of terms such as high or low system sensitivity is rather meaningless, unless a well-defined quantitative measure of this sensitivity is being referred to.

The sensitivity vector function, $\underline{\sigma}(t)$, can be used to define a general quadratic index of the system sensitivity over a specified time interval, due to changes in a single design parameter:

$$J_s = \int_0^T \underline{\sigma}^T \, \underline{S} \, \underline{\sigma} \, dt \tag{2.7}$$

where $\underline{S}$ is an arbitrary positive semi-definite weighting matrix, which can be chosen consistent with the overall system requirements. When two or more simultaneous parameter variations are considered, a sensitivity function corresponding to each parameter must be computed. A general sensitivity index can then, for example, be defined in terms of the extended sensitivity vector:

$$\underline{\tilde{\sigma}}^T(t) = [\, \underline{\sigma}_1^T \,,\, \underline{\sigma}_2^T \,,\, \ldots,\, \underline{\sigma}_k^T(t) \,] \tag{2.8}$$

which is an n•k dimensional vector containing all the elements of the sensitivity vector functions corresponding to k parameter variations. The sensitivity index then becomes:

$$J_s = \int_0^T \underline{\tilde{\sigma}}^T \, \underline{\tilde{S}} \, \underline{\tilde{\sigma}} \, dt \tag{2.9}$$

where $\tilde{\underline{S}}$ is an $(n \cdot k) \times (n \cdot k)$ positive semi-definite matrix.

In order to compute the value of J_s as expressed by Equation (2.9), it is necessary to solve Equation (2.5) k times, i.e. once for each parameter variation. This difficulty can be avoided by using the first order variation of the state in defining the sensitivity index, instead of the sensitivity functions. The choice of the weighting matrix is also simplified in this case. This index can be written in the form:

$$J_s = \int_0^T \delta \underline{y}^T \, \underline{S} \, \delta \underline{y} \; dt \tag{2.10}$$

where $\underline{S}$ is an $n \times n$ weighting matrix and $\delta \underline{y}(t)$ is the deviation of the state due to a specified variation of all the variable parameters.

A sensitivity index of this form has the disadvantage of depending on the actual deviation of the system state rather than its derivatives. Consequently it must be kept in mind that J_s as defined by Equation (2.10) depends on the specified parameter variation.

The equation describing $\delta \underline{y}(t)$ is obtained by taking the first order variation of Equation (2.2):

$$\delta \dot{\underline{y}}(t) = \underline{A} \, \delta \underline{y}(t) + \delta \underline{A} \, \underline{y}(t) + \delta \underline{c} \, u(t) \tag{2.11}$$

where $u(t)$ is assumed to be an external input, which is independent of the parameter variations. Thus, only one differential equation must be solved in order to compute the value of J_s, as given by Equation (2.10), regardless of the number of parameter variations involved.

The assumption here is, that a representative set of parameter variations can be determined for the computation of $\delta \underline{y}(t)$. As an example, worst case conditions could be used for this purpose. This may appear to be a limitation of the sensitivity index defined by Equation (2.10), but some judgment of the relative variations of these parameters would also have to be made when choosing the $\underset{\sim}{S}$ weighting matrix in Equation (2.9). When only a single parameter variation is considered the sensitivity indices of Equations (2.9) and (2.10) are actually equivalent, since in this case:

$$\delta \underline{y}(t) = \underline{\sigma}(t) \frac{\delta \xi}{\xi} \tag{2.12}$$

In many instances, the variations of the parameters can be described by a statistical distribution, in which case the expected value of the sensitivity index may be used:

$$\bar{J}_s = \int_0^T \overline{\delta \underline{y}^T \underline{S} \, \delta \underline{y}} \, dt \tag{2.13}$$

where the bar indicates the ensemble average of the quantity. A sensitivity index of this type was proposed by Mazer [24], whose definition is given in terms of the first component of $\delta \underline{y}(t)$, which represents the variation of the output response.

<u>Example 2.1</u>

The characteristics of a sensitivity index of the type given by Equation (2.9) or (2.10) is demonstrated by a simple example. Consider the third order closed-loop system shown in Figure 2.2.

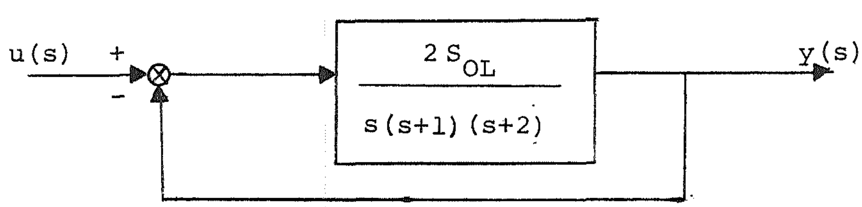


Figure 2.2 Third order system

The open-loop static sensitivity, S_{OL}, is taken to be the variable parameter. S_{OL} is also the free design parameter, i.e. its nominal value can be chosen by the designer. The sensitivity function of the output is then given by:

$$\sigma_1(t) = \frac{dy(t)}{dS_{OL}/S_{OL}} \tag{2.13}$$

which can be used in a sensitivity index of the form:

$$J_S = \int_0^{\infty} \sigma_1^2(t)\ dt \tag{2.14}$$

The input is the unit step function and the integral is taken from t=0 to infinity. This integral has a finite value as long as the system response is stable, since the close-loop static sensitivity is always equal to unity and, therefore, $\lim_{t\to\infty} \sigma_1(t) = 0$.

The value of the sensitivity index was computed as a function of the free design parameter, S_{OL}, as shown in Figure 2.3 for a range of values, which give a stable response. Thus, it is seen that this system has minimum sensitivity to changes in the open-loop static sensitivity, as defined by J_s, when $S_{OL} \simeq 0.4$, which corresponds to a damping ratio of $\zeta \simeq 0.75$ of the dominant second order mode. The sensitivity index of Equation (2.14), therefore, defines an absolute minimum sensitivity of this system with respect to the open-loop static sensitivity. If the system specifications were only concerned with the deviations of the output due to changes in S_{OL}, this value of S_{OL} would seem to be a reasonable choice.

The sensitivity function for three values of S_{OL} is shown in Figure 2.4. It is interesting to note, that the maximum value of $\sigma_1(t)$, which is proportional to the maximum deviation of the system output,

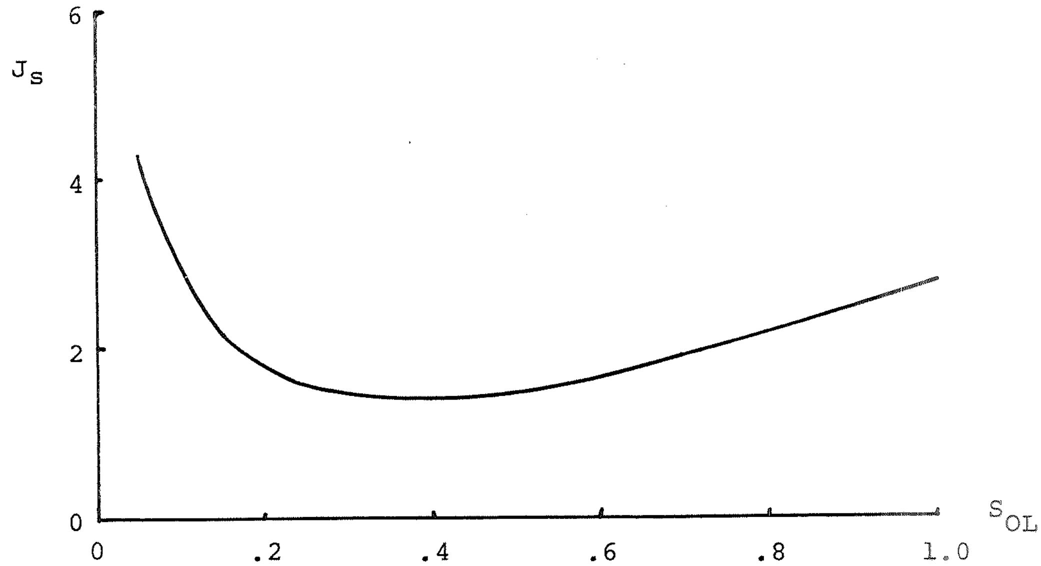


Figure 2.3. Sensitivity index as a function of S_{OL}

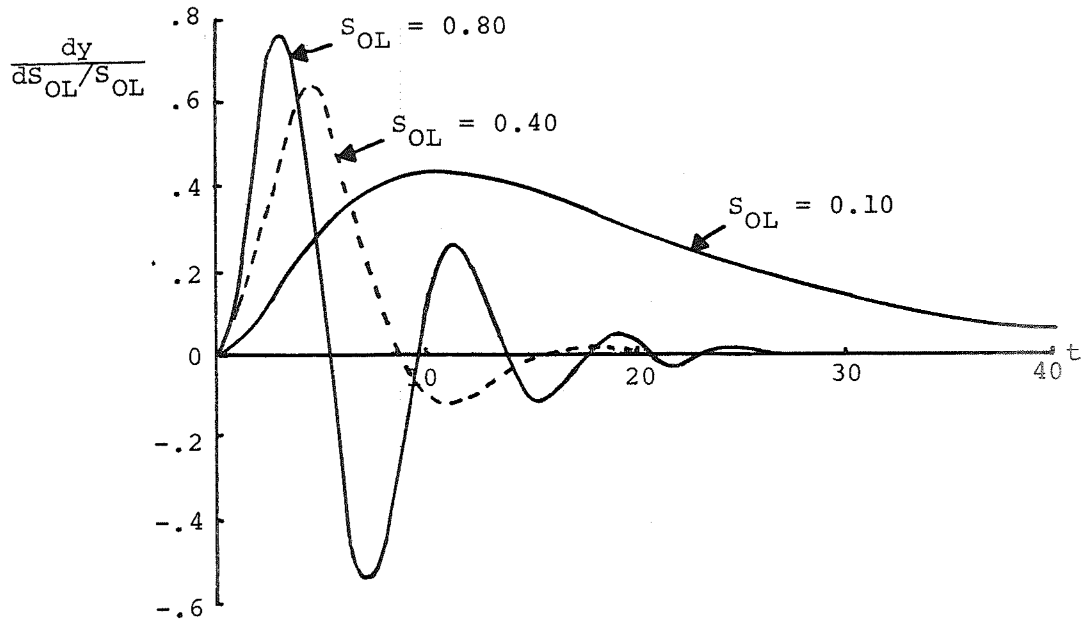


Figure 2.4. Sensitivity functions w.r.t. S_{OL}

increases monotonically as a function of S_{OL}. Hence, if $\sigma_{1_{max}}$ were to be used as an index of sensitivity, the absolute minimum sensitivity would correspond to $S_{OL}=0$, which is not a very meaningful result. For $S_{OL}=.4$, $\sigma_1(t)$ is seen to combine a relatively small maximum value with a fast settling time.

In most applications, however, system sensitivity must be considered in the context of specific requirements on the system response. The sensitivity index of the type discussed above is potentially useful in this case as a component of the performance index which expresses the overall system performance.

2.3 Transfer Function Sensitivity

In addition to time domain sensitivity, two other definitions of system sensitivity have been found useful in control system design. These are Bode's [1] definition of the transfer function sensitivity, and the sensitivity of the closed-loop poles and zeros to changes in the open-loop system characteristics.

The sensitivity of the closed-loop transfer function of the system shown in Figure (2.1), due to changes in a system design parameter, can be defined as:

$$S_\xi^G = \frac{dG/G}{d\xi/\xi} \tag{2.15}$$

where G is the closed-loop transfer function given by:

$$G = \frac{G_1 P}{1+G_1 G_2 P} \tag{2.16}$$

Depending on the location of the parameter variation, S_{ξ}^{G} can be written as:

$$S_{\xi}^{G} = \frac{dG/G}{dP/P} \cdot \frac{dP/P}{d\xi/\xi} = S_{p}^{G}\, S_{\xi}^{P}$$

$$S_{\xi}^{G} = \frac{dG/G}{dG_1/G_1} \cdot \frac{dG_1/G_1}{d\xi/\xi} = S_{G_1}^{G}\, S_{\xi}^{G_1} \qquad (2.17)$$

$$S_{\xi}^{G} = \frac{dG/G}{dG_2/G_2} \cdot \frac{dG_2/G_2}{d\xi/\xi} = S_{G_2}^{G}\, S_{\xi}^{G_2}$$

where the three equations correspond to parameter variations in P, G_1 and G_2, respectively. S_{ξ}^{P}, $S_{\xi}^{G_1}$ and $S_{\xi}^{G_2}$ represent the effects of unit parameter variations on their respective open-loop transfer functions. S_{P}^{G}, $S_{G_1}^{G}$ and $S_{G_2}^{G}$ relate the changes in these open-loop transfer functions to the change in the closed-loop transfer function. By differentiation of Equation (2.16) with respect to P, G_1 and G_2 it is easily shown that:

$$S_{P}^{G} = S_{G_1}^{G} = \frac{1}{1+G_1G_2P} \qquad (2.18)$$

and

$$S_{G_2}^{P} = -G_2G \qquad (2.19)$$

A relationship between these transfer function sensitivities and the sensitivity function of the system output response can be obtained by considering the transform of the system output, which is given by:

$$y(s) = G(s)\ u(s) \qquad (2.20)$$

Differentiating this equation with respect to the variable system parameter and using the definition of the sensitivity function of the output, the following expression is obtained:

$$\sigma(s) = \frac{dy(s)}{d\xi/\xi} = S_\xi^G \, y(s) \tag{2.21}$$

Thus, S_ξ^G is a transfer function, that relates the sensitivity function to the corresponding output response. If it is assumed that the system output response is completely specified, then the output sensitivity function can only be controlled by adjusting S_ξ^G. Horowitz[17] has shown, that in the case of changes in the plant, S_ξ^G can be adjusted independent of the closed-loop transfer function only when both G_1 and G_2 can be chosen by the designer. In this case Equation (2.21) can be written:

$$\sigma(s) = \frac{S_\xi^P}{1+G_1G_2P} \, y(s) \tag{2.22}$$

Since y(s) is specified and the changes in P as represented by S_ξ^P are, in most cases, not under the control of the system designer, only the total open-loop transfer function, $G_{OL} = G_1G_2P$, is available for adjusting the sensitivity response. Hence, the open-loop gain would be chosen as large as possible in the frequency band of the system output in order to reduce the output sensitivity. This can be achieved by using large feedback gain at these frequencies. The forward loop compensation, $G_1(s)$, is then appropriately chosen in order to achieve the specified closed-loop transfer function.

From Equation (2.18) it can be seen that changes, which occur in the forward loop compensation, have the same effect on system output as if they had taken place in the plant. In this case:

$$\sigma(s) = \frac{S_{\xi}^{G_1}}{1+G_1G_2P} \quad y(s) \tag{2.23}$$

The difference here is, that some control of $S_{\xi}^{G_1}$ and the corresponding parameter variations may be possible. Similarly, the output sensitivity to changes in the feedback path can be written:

$$\sigma(s) = -G\ G_2\ S_{\xi}^{G_2}\ y(s) \tag{2.24}$$

It was found desirable to use large values of G_2 in order to reduce the output sensitivity to changes in the forward loop elements. Equation (2.24) shows, that this has the effect of increasing the output sensitivity to changes in the feedback path. In order to achieve overall reduction in the sensitivity of the system output in this way, it is necessary to require the variations in the feedback compensation to be small. This point is sometimes all but neglected in the discussion of control system sensitivity.

Another undesirable effect of large feedback gain is its amplification of noisy signals. Consider, for instance, the effect of sensor noise, which enters the system at the feedback level as shown in Figure 2.5.

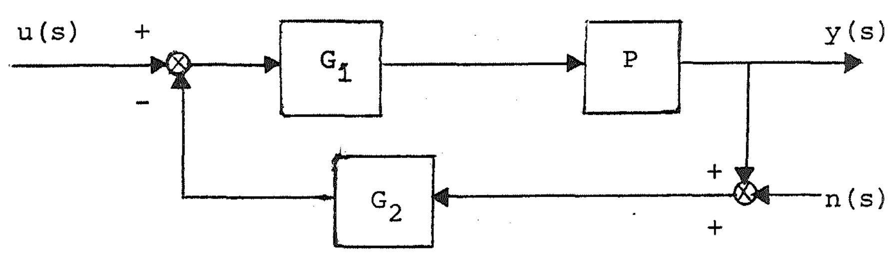


Figure 2.5 System with feedback noise

The effect of the noise on the output is expressed by the transfer function:

$$\frac{y(s)}{n(s)} = -\frac{G_1G_2P}{1+G_1G_2P} \tag{2.25}$$

Thus, if the open-loop transfer function, G_1G_2P, is large in the frequency band of n(s), there is very little attenuation of the noise. Hence, it may be necessary to restrict the bandwidth of the feedback elements which may conflict with the goal of decreasing system sensitivity.

A sensitivity index in terms of S_ξ^G $(j\omega)$ can be obtained, for example, by writing Equation (2.14) in the form:

$$J_s = \int_0^\infty \sigma^2(t)\ dt = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} S_\xi^G(j\omega)S_\xi^G(-j\omega)y(j\omega)y(-j\omega)d(j\omega) \tag{2.26}$$

where Equation (2.21) has been used to substitute for $\sigma(s)$. This form of the sensitivity index may be useful when the transform of the system output is specified, since J_s is now an explicit function of y(s).

In the discussion of control system sensitivity it has been found convenient to require the same nominal response to be maintained in order to give a basis for comparison of designs with different means of compensation. By so doing, the problem of system sensitivity is separated from that of achieving the desirable response characteristics. This is a matter of convenience and may not always be possible or desirable as, for example, when the designer has limited freedom in choosing the form and location of the compensating elements.

2.4 Closed-loop Pole Sensitivity

Closed-loop pole sensitivity has been found to be a useful tool for analyzing the effects of changes in the open-loop parameters on the closed-loop response characteristics of the system. This is especially true for systems with relatively few dominant modes, as is the case in most flight control systems. The sensitivities of the i^{th} closed-loop pole, p_i, due to changes in the open-loop static sensitivity, poles and zeros, are defined as follows:

$$
\begin{aligned}
S^i_{S_{OL}} &= \frac{\partial p_i}{\partial S_{OL}/S_{OL}} \\
S^i_{\tilde{p}_j} &= \frac{\partial p_i}{\partial \tilde{p}_j} \\
S^i_{\tilde{z}_j} &= \frac{\partial p_i}{\partial \tilde{z}_j}
\end{aligned}
\tag{2.27}
$$

where S_{OL} is the open-loop static sensitivity, and $\tilde{p}_j$ and $\tilde{z}_j$ denote the j^{th} open-loop pole and zero, respectively. The following expressions have been derived for these sensitivities in Reference[26] for simple closed-loop poles:

$$
\begin{aligned}
S^i_{S_{OL}} &= -\left[\frac{(s-p_i)\ G_{OL}}{1 + G_{OL}}\right]_{p=p_i} \\
S^i_{\tilde{p}_j} &= \frac{p_i}{\tilde{p}_j}\ \frac{S^i_{S_{OL}}}{(p_i-\tilde{p}_j)} \\
S^i_{\tilde{z}_j} &= \frac{p_i}{\tilde{z}_j}\ \frac{S^i_{S_{OL}}}{(\tilde{z}_j-p_i)}
\end{aligned}
\tag{2.28}
$$

A somewhat simpler derivation of these equations is given in Appendix A of this thesis.

The closed-loop pole sensitivities, are, in general, complex numbers which can be used to compute the first order variations of any closed-loop pole due to changes in the open-loop parameters. The effect of such a variation on the time response depends on the nominal location of the corresponding closed-loop pole. These sensitivities are, therefore, most useful when this location in the complex plane is specified. Then it is only necessary to determine if the poles move too far from these locations to adversely affect the response. The sensitivity of closed-loop zeros to variations in the open-loop parameters is easily determined, since the closed-loop zeros consist of the forward path open-loop zeros, as well as the open-loop poles of the feedback path.

It is of interest to determine the relationship between the closed-loop pole sensitivities and the output sensitivity function, $\sigma(t)$. Assuming, for example, that only the closed-loop poles are affected by a parameter change, the following relationship can be derived:

$$\sigma(s) = \frac{dy(s)}{d\xi/\xi} = \frac{dG}{d\xi/\xi} u(s) = \left[\sum_{i=1}^{n} \frac{\partial G}{\partial p_i} \frac{\partial p_i}{\partial \xi/\xi}\right] u(s) \tag{2.29}$$

But $\frac{\partial G}{\partial p_i}$ can be written as:

$$\frac{\partial G}{\partial p_i} = \frac{\partial}{\partial p_i}\left[\frac{S_{OL} \prod_{i=1}^{m} (1-\frac{s}{z_i})}{\prod_{j=1}^{n} (1-\frac{s}{p_j})}\right] = \frac{s}{p_i} \frac{G}{(s-p_i)} \tag{2.30}$$

where p_i is assumed to be a distinct pole. If the variable parameter is taken to be an open-loop pole, $\tilde{p}_j$, the following expression for

$\sigma(s)$ is obtained, using Equations (2.27), (2.29) and (2.30):

$$\sigma(s) = \left[\sum_{i=1}^{n} \frac{\tilde{p}_j}{p_i} \frac{S^i_{\tilde{p}_j}}{(s-p_i)} \right] s\; y(s) \tag{2.31}$$

The sensitivity function can, therefore, be obtained as the summed output of n parallel first order filters driven by the system output, as shown in Figure 2.6. The poles of these filters are the system closed-loop poles and their gains are proportional to the closed-loop pole sensitivities. Thus each of these filters produces the contribution of a specific closed-loop pole to the output sensitivity function corresponding to an open-loop pole. Similar relationships can be obtained when the variable parameter is an open-loop zero or the open-loop static sensitivity. Thus, for a variation of the open-loop zero, $\tilde{z}_j$:

$$\sigma(s) = \left[\sum_{i=1}^{n} \frac{\tilde{z}_j}{p_i} \frac{S^i_{\tilde{z}_j}}{(s-p_i)} \right] s\; y(s) \tag{2.32}$$

and in the case of the open-loop static sensitivity, S_{OL}:

$$\sigma(s) = \left[\sum_{i=1}^{n} \frac{S_{OL}}{p_i} \frac{S^i_{SOL}}{(s-p_i)} \right] s\; y(s) \tag{2.33}$$

where the closed-loop static sensitivity, S_{CL}, is assumed unaffected.

It is clear from Figure 2.6, that for a system with specified closed-loop poles, the only way in which the output sensitivity function can be influenced is through the closed-loop pole sensitivities. Hence, it may be expected that the minimization of some

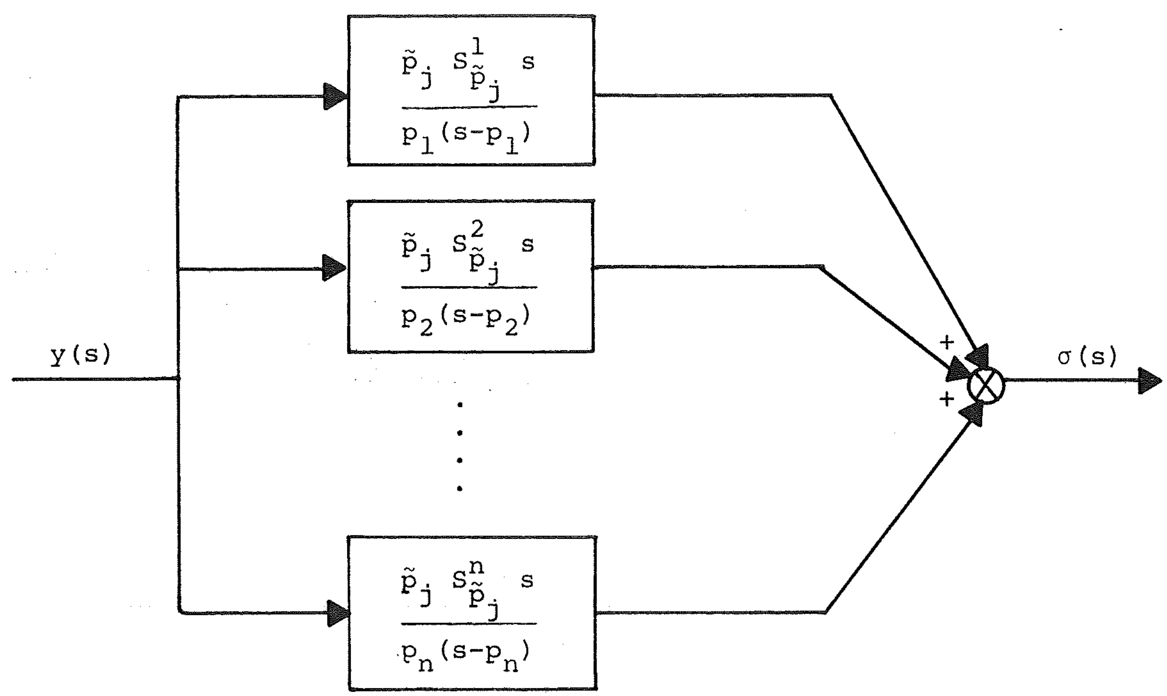


Figure 2.6. Effect of closed-loop pole sensitivities on the output sensitivity function.

sensitivity index, defined in terms of $\sigma(t)$ and its derivatives, leads to a reduction of the corresponding closed-loop pole sensitivities. The relative reduction of a closed-loop pole sensitivity is likely to depend on the importance of the corresponding mode in the system response.

Finally, Equations (2.21) and (2.31) can be used to relate the sensitivity of the transfer function to the closed-loop pole sensitivity due to an open-loop pole variation, which is assumed to affect only the closed-loop poles:

$$S^{G}_{\tilde{p}_j}(s) = s\left[\sum_{i=1}^{n} \frac{\tilde{p}_j}{p_i} \frac{S^{i}_{\tilde{p}_j}}{(s-p_i)}\right] \tag{2.34}$$

Similar expressions for the other open-loop parameters are readily obtained from Equations (2.32) and (2.33).

2.5 System Specifications

In a practical control system design the question of system sensitivity must be considered as it relates to the overall performance requirements. The definition of sensitivity which is most useful depends to some extent on the form in which these requirements are expressed. For this reason, and for the purpose of later development, it is appropriate to consider the types of system specifications which are commonly used in control system design.

2.5.1 Time Domain Specifications

One form of system specifications which has been used very extensively is the envelope of the system step response of the type shown in Figure 2.7, which specifies the permissible tolerances of the system output due to a unit step input.

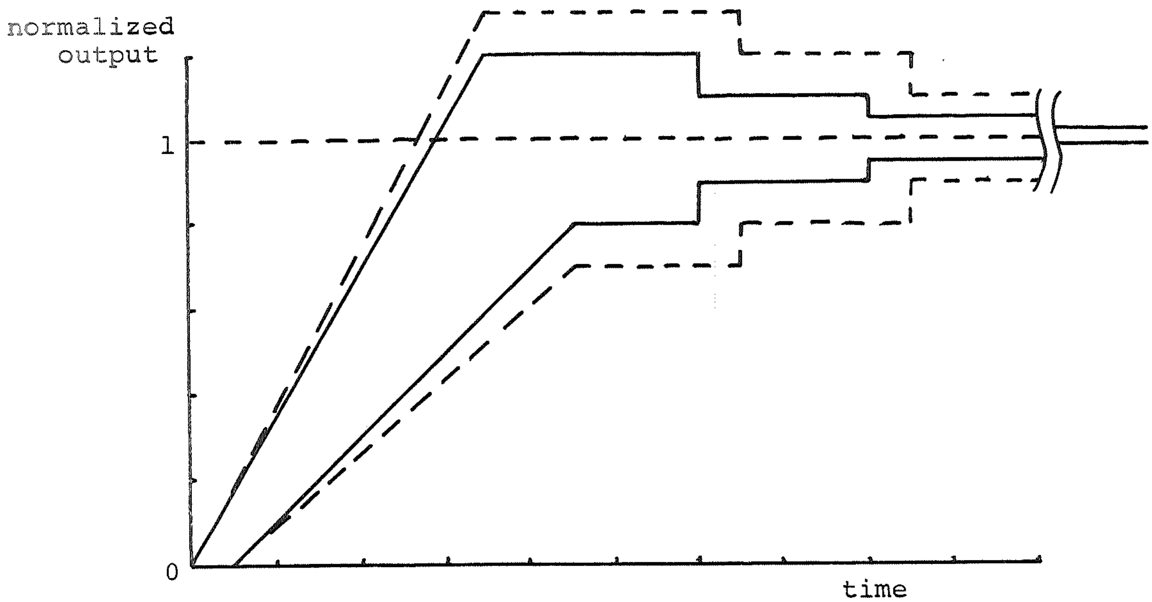


Figure 2.7. Normalized step response specifications

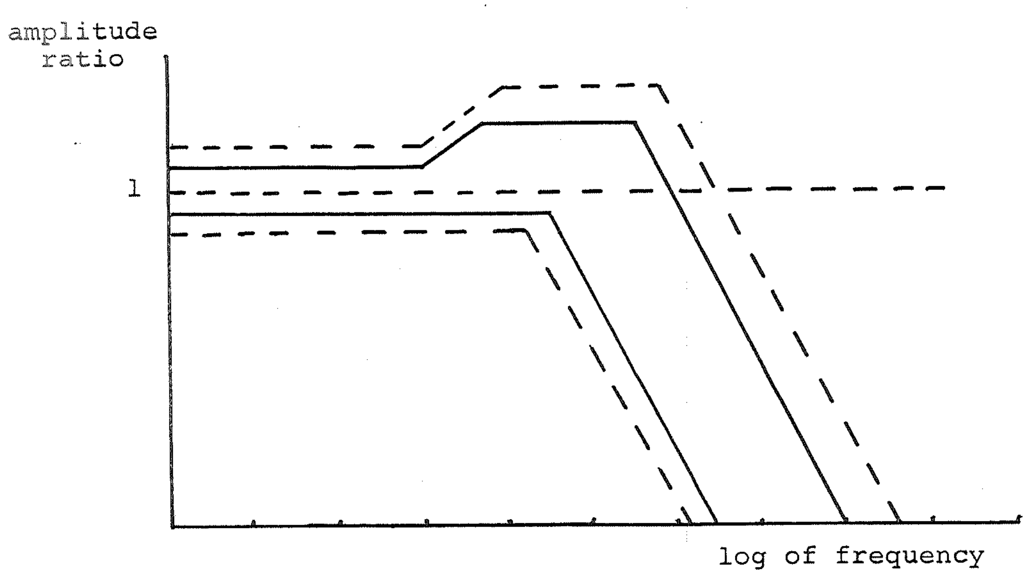


Figure 2.8. Frequency response specifications

The implication is that any system design which has a step response lying within the specified envelope is satisfactory. Thus any such response is equally acceptable regardless of how close to or far from the boundaries it lies. When a fixed set of system parameters is considered it is, therefore, sufficient to determine a compensation that results in a satisfactory system response for the given set of design parameter values. If, on the other hand, some of the system parameters vary over a range of values, or are not accurately known, the problem becomes more difficult. It is no longer sufficient that the design, based on the nominal parameter values, have a satisfactory response.

The output response may now be required to remain within the envelope for all possible parameter variations. A different statement of this specification was suggested in Reference [44] for systems, whose parameter values are known in terms of their statistical distribution. In this case the design may be required to satisfy the specifications with a stated probability. Thus, any design which satisfied the response specifications with a probability equal to or better than the specified probability would be acceptable. Alternatively, more than a single response envelope could be specified as shown in Figure 2.7. Varying degrees of acceptability can then be associated with each envelope or the nominal response may be required to satisfy the most constraining one with less severe requirements on the off-nominal response.

The question arises how specifications of this type can be interpreted in terms of requirements on the sensitivity of the system response. Given any nominal response which satisfies the specifications, it is clear that any deviations due to parameter variations must be bounded if the response is to remain within the tolerances. The requirements on the sensitivity of the output, therefore, depend on both the nominal response and the tolerances as expressed by the response envelope. Hence, it is a matter of convenience to prescribe the nominal response and deal with the sensitivity problem separately. As has been pointed out, this requires a great amount of freedom in the design, which may not always be available.

In any event, it is necessary that the sensitivity of the time response be bounded if the specifications are to be met. The sensitivity indices discussed in Section 2.2.1 are measures of the magnitude of the response sensitivity. The effect of using the quadratic form in these indices is to emphasize the peaks of the sensitivity function which correspond to the peak deviations of the response. It may be expected, that by controlling the value of a sensitivity index of this type, the actual deviations of the system time response can be constrained.

2.5.2 Frequency Response Specifications

The frequency response can be specified by its tolerances at all frequencies of interest analogous to the specifications in the time domain. This results in an envelope of all acceptable frequency responses of the type shown in Figure 2.8. When considering a system with uncertain design parameters, the design may be required to satisfy these specifications with a stated probability or varying degrees of acceptability may be indicated as shown. The relationship of the transfer function sensitivity to these specifications is analogous to the relationship between the sensitivity function and the time response envelope. Thus, given a nominal frequency response of the system, the sensitivity of the transfer function must be limited if the specifications are to be met for all possible parameter variations.

When non-minimum phase systems are considered, it is necessary to specify the tolerances of the phase-angle as a function of frequency in addition to the amplitude response. These specifications can, in theory, be interpreted in terms of specifications on the system time response.

2.5.3 Complex Plane Specifications

Specifications of the acceptable closed-loop root locations is a convenient method for defining the satisfactory response of a system with relatively few dominant modes. These locations may be given as bounded areas in the s-plane, which correspond to the time or frequency response envelopes. Thus, in the case of variable design parameters, the closed-loop roots could be required to remain within the assigned areas with a given probability. The relationship of the closed-loop root sensitivities to these specifications is again dependent on the nominal system design. Given the nominal locations of the closed-loop roots, their sensitivities to the appropriate parameter variations must be small enough so that the roots remain within the specified areas in the complex plane.

2.6 Sensitivity Design Methods

A number of methods have been proposed for designing control systems, which satisfy specifications of the type discussed in Section 2.5 despite changes in some system parameters. Although the emphasis in this report is on time domain methods, it is of interest to review the frequency domain and complex plane approaches, which can be useful in selecting the type of compensation when the configuration of the controller must be chosen.

2.6.1 Time-Domain Design

Time-domain design methods commonly require the performance of the system to be expressed in terms of an index, which is a function of the system time response. This performance index can sometimes be regarded as a measure of how well the system is doing relative to a desired performance, which presumably satisfies all the system specifications and is achieved when the value of the performance index goes to zero. The optimum design, relative to the desired performance, is obtained by minimizing the performance index with respect to the available control variables.

When considering the design of a system with uncertain or variable design parameters, the effects of the uncertainties on the system performance may be taken into account by somehow including these effects in the performance index of the system in an attempt to satisfy all the system specifications simultaneously. This can be done, for instance, by adding a sensitivity index of the type discussed in Section 2.2.1 to the index representing the nominal system performance. The assumption is, that minimization of this expanded performance index will reduce the effects of the parameter variations on the system response, in addition to obtaining a desired nominal response.

The design problems are usually divided into two categories:

— free configuration or optimal control
— fixed configuraiton or parameter optimization

Considerable effort [7], [22], [36] has been spent on studying the optimization of a quadratic performance index of the form:

$$J = \int_0^T (\underline{x}^T \underline{Q}\, \underline{x} + \underline{\sigma}^T \underline{S}\, \underline{\sigma} + \underline{u}^T \underline{R}\, \underline{u})\, dt \tag{2.35}$$

where $\underline{x}(t)$ is the state vector of a linear system with a given initial condition, $\underline{\sigma}(t)$ is the sensitivity function of the state with respect to a single variable parameter and $\underline{u}(t)$ is to be determined so that J is minimized over the time interval (0,T). The system variables are described by equations of the form:

$$\dot{\underline{x}} = \underline{A}\, \underline{x} + \underline{B}\, \underline{u} \quad ; \quad \underline{x}(0) = \underline{x}_0 \tag{2.36}$$

$$\dot{\underline{\sigma}} = \underline{A}\, \underline{\sigma} + \frac{\partial \underline{A}}{\partial \xi} \underline{x} + \frac{\partial \underline{B}}{\partial \xi} u + \underline{B} \frac{\partial \underline{u}}{\partial \xi} \quad ; \quad \underline{\sigma}(0) = \underline{0} \tag{2.37}$$

There are several drawbacks to this approach. First, the sensitivity function cannot be determined unless the functional relationship of the control vector, $\underline{u}(t)$, to the state is known. The difficulty is usually avoided by defining $\underline{u}(t)$ as a linear function of $\underline{x}$ and $\underline{\sigma}$, leaving the feedback gains free to be selected. This makes it possible to obtain explicit equations for $\underline{\sigma}$, but $\underline{u}$ has now become a function of variables which are unavailable in any form in the system. The controller must then compute the solutions of $\underline{\sigma}(t)$ in real time, which is likely to be an undesirable complexity. In addition, the improvement in the sensitivity of the response due to the feedback of $\underline{\sigma}$ may be insignificant and has been found to have a deteriorating effect in some instances[36].

When multiple parameter variations are considered, an equal number of sensitivity functions must be added to the performance index. Since Equation (2.37) must be solved separately for each variable parameter, the computational task becomes prohibitive. Consequently, only single parameter variations have been assumed in most of the studies of the subject. If $\underline{u}$ is taken to be a function of $\underline{x}$ only, the structure of the controller is simplified, since it is no longer necessary to compute $\underline{\sigma}$ in real time. The minimization of the performance index for multiple parameter variations is, however, still a difficult task. The choice of weighting matrices for the sensitivity index is an area of some ambiguity, since no systematic method has been proposed for making this choice.

A somewhat different approach to the problem of parameter uncertainty was taken by Tuel[43] who defined the system performance index, knowing the statistical distribution of the parameters as the expected value of the quadratic performance index:

$$\bar{J} = \int_0^T \overline{(\underline{x}^T \underline{Q}\, \underline{x} + \underline{u}^T \underline{R}\, \underline{u})}\, dt \tag{2.38}$$

This formulation of the problem is not limited to small parameter variations, but the control law which minimizes $\bar{J}$ cannot, in general, be put into a feedback form.

The parameter optimization method would appear to be more promising for a practical control system design including system sensitivity. The reason is, that many of the analytical and computational difficulties, which are associated with the optimal control design, can be avoided by specifying the configuration of the system. Such a method was developed by Mazer[24] based on the minimization of the mean square value of the system output deviations with respect to designated system parameters,using a periodic input and specifying the nominal response of the system. The computational difficulties in obtaining the optimum parameter values are considerable, however, for all but low order systems since the mean square value must be evaluated, using Parseval's theorem, as an integral over all frequencies.

No method of the types discussed can guarantee that specifications, such as those of Section 2.5.1, will be satisfied. Thus, a relatively low sensitivity design may not be acceptable unless its nominal response is also suitably situated with respect to its tolerance envelope. Conversely, a good nominal response is acceptable only if the deviations, due to the specified parameter variations, do not violate the appropriate boundaries. Hence, the minimization of the performance index is a useful design tool, but the acceptability of the solution must be judged on the basis of how well it satisfies the original system specifications.

2.6.2 Frequency Domain Design

The most successful methods for reducing the effects of plant variations on the system performance have been formulated in terms of the frequency response. Horowitz[17] developed a method whereby the open- and closed-loop transfer functions can be determined such as to satisfy specifications of the type discussed in Section 2.5.2, assuming that the compensation in the forward and feedback paths can be chosen freely. The open-loop transfer function, G_{OL}, uniquely determines the sensitivity of the closed-loop frequency response to changes in the forward path as seen from Equation (2.18). The advantage of this technique is that it is not limited to small parameter variations and it is directly related to the specifications on the frequency response. This is seen from the fact that the ratio of the closed-loop response and its nominal value can be expressed as:

$$\frac{G(j\omega)}{G_*(j\omega)} = \frac{1 + G^*_{OL}(j\omega)}{\frac{P(j\omega)}{P_*(j\omega)} + G^*_{OL}(j\omega)} \qquad (2.39)$$

where P represents any off-nominal frequency response of the plant. Specifications of the type shown in Figure 2.8 can be used to determine the extreme values of $|\frac{G}{G_*}|$ for all frequencies of interest. Knowing the extreme values of $\frac{P}{P_*}$, G^*_{OL} can then be determined so that the ratio of the closed-loop responses remains within the specified limits. It then remains to select the proper compensation for realizing G^*_{OL}, which must be distributed between the forward and feedback paths in such a way that the desired closed-loop response, G_*, is achieved.

This method can be used to deal with large changes in the transfer function of the plant, which need be known only in terms of the extremes of the frequency response. It is, however, not very suitable for taking into account changes in the characteristics of the compensation, whose elements are determined after a suitable open-loop frequency response has been found. In the case of non-minimum phase systems, both amplitude and phase response specifications must be considered when determining the satisfactory open-loop frequency response.

2.6.3 Complex Plane Design

The objective of the complex plane methods is to ensure that the dominant closed-loop roots remain within specified areas in the complex plane for all possible operating conditions. This means, for instance, that the movement of the closed-loop poles must somehow be restricted despite changes in their open-loop counterparts or the loop gain. A straightforward method for achieving this goal is to place compensating zeros close to the desired pole locations, which in conjunction with high loop gain ensures that the closed-loop poles will be close to the zero locations regardless of open-loop changes. The assumption is then, that the zeros of the compensation are highly stable which is a part of the price paid for low sensitivity to changes in the plant.

If these zeros are not to alter the response characteristics, they must be placed in the feedback path of the single-loop system, which eliminates them from the closed-loop transfer function. The accompanying poles must be at high enough frequency such as to have a small effect on the system response. The dominant system poles are, therefore, stabilized by increasing the gain of the feedback path in the frequency band of the corresponding dominant mode.

Cancellation of the varying open-loop pole may also be attempted by locating a zero in the forward path. The associated pole is then placed in some desirable location. Again, the compensation must be quite stable and the gain must be large enough for an effective cancellation despite changes in the pole location. Variable open-loop zeros can be dealt with in a similar way by placing a pole in its vicinity, either in the forward or feedback paths. This approach to the sensitivity problem is useful for determining the type of compensation to be used in a fixed configuration design in the time domain.

The complex plane design is mainly concerned with the low-frequency dominant modes and constrains the higher frequency modes to be well damped in order to have a small effect on the system response. Thus, the sensitivity problem is to some extent separated from the stability problem, which is considered as a constraint on the design. This is a convenient approach, but could conceivably result in an unnecessarily complicated design.

Techniques for determining the locations of the compensation singularities have been developed[16][20] for satisfying specifications of the type discussed in Section 2.5.3.

One of the characteristics of the sensitivity methods, which have been reviewed in this section, is that they are considerably more complicated than similar methods which assume that all the design parameters are invariant. In the case of frequency domain methods the problem of system sensitivity is often separated from the question of system stability and the achievement of a desirable nominal response. This is a matter of convenience which often requires a great amount of freedom in the choice of compensation.

The application of time domain methods suffers from the ambiguity introduced by the arbitrary choice of the sensitivity weighting matrix. Typically, these methods require the feedback of all system variables, including the sensitivity functions which must be computed in real time. The resulting design may be undesirable or impractical because of the associated complexity. The numerical difficulties in obtaining the solution have limited the application of these methods to relatively simple problems.

A time domain method, which alleviates some of these difficulties, will be developed in Chapter 3.

Chapter 3. A Sensitivity Design Method

3.1 Introduction

In this chapter, a method will be developed for the design of a linear control system in which some of the system parameters are known only in terms of their statistical distribution. When the design problem is defined in terms of a specific performance index, a straightforward approach consists of minimizing the statistical expectation of the performance index using the known parameter distribution and subject to the constraints imposed by the system dynamics. The task of obtaining a solution to this problem is very difficult, in general, and some simplifying assumptions are necessary. First it is assumed that a quadratic performance index expressed in terms of the system state will be used. The state is extended to include the state of a reference model. Furthermore, the choice of system configuration will be made a priori by the designer with certain designated free design parameters whose values can be chosen to minimize the expectation of the performance index. The fixed configuration allows the designer to restrict the complexity of the system in advance, which has been found to be a very practical design approach. In addition, one avoids the problems associated with determining the variation of the system state due to parameter changes, when the form of the control feedback configuration has not yet been determined. The phase-variable form of the state equations will be used throughout this thesis because of significant computational advantages. These equations are also convenient for transforming the system transfer functions into state equations and vice versa. This is especially the case when studying the effects of different methods of compensation. The design method will be developed for a single input/output system with the unit step function as the standard input. Extension to multivariable systems is straightforward but increases the computational task considerably.

Finally, it will be assumed that the parameter variations are such that the resulting change in the state vector is sufficiently described by its first order variation.

3.2 System Equations

One possible state-space realization of a general transfer function:

$$\frac{y(s)}{u(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0} \qquad (m<n) \tag{3.1}$$

was given in Section 2.2.2 as:

$$\dot{\underline{y}}(t) = \underline{A}\,\underline{y}(t) + \underline{c}\,u(t) \quad ; \quad \underline{y}(0) = \underline{0} \tag{3.2}$$

where $\underline{y}$ is an n-dimensional state vector whose first element, y_1, is the system output, $\underline{A}$ is an n×n matrix in the phase-variable form:

$$\underline{A} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 1 \\ -a_0 & -a_1 & -a_2 & \ldots & -a_{n-1} \end{bmatrix}$$

and $\underline{c}$ is given by Equation (2.3). For a system whose output reaches a steady-state as time goes to infinity, it is convenient to use only the transient part of the response. This is obtained by subtracting the steady-state value from the state vector. Thus:

$$\underline{x}(t) = \underline{y}(t) - \underline{y}_{ss} \tag{3.3}$$

where $\underline{x}(t)$ is the transient response and:

$$\underline{y}_{ss} = \lim_{t\to\infty} \underline{y}(t) \tag{3.4}$$

For a unit step input the equation for $\underline{x}(t)$ can be written in the homogeneous first order form:

$$\dot{\underline{x}}(t) = \underline{A}\ \underline{x}(t) \quad ; \quad \underline{x}(0) = \underline{x}_0 \tag{3.5}$$

where the initial condition includes the effects of the system input. The appropriate form of this initial condition can be determined by finding the steady-state value of the state from Equation (3.2) by setting the derivative equal to zero:

$$\underline{A}\ \underline{y}_{ss} + \underline{c} = \underline{0} \tag{3.6}$$

since $u(t) = 1$ for $t \geq 0$. Using the phase-variable form of the system matrix, this equation can be written as:

$$y_{(i+1)_{ss}} = -c_i \qquad 1 \leq i \leq n-1 \tag{3.7a}$$

$$\sum_{i=1}^{n} a_{i-1}\, y_{i_{ss}} = c_n \tag{3.7b}$$

By using Equation (2.3) for c_n and substituting from Equation (3.7a), $y_{1_{ss}}$ is obtained from (3.7b):

$$a_0 y_{1_{ss}} - \sum_{i=n-m}^{n-1} a_i\, c_i = b_0 - \sum_{i=n-m}^{n-1} a_i\, c_i \tag{3.8}$$

since $c_i=0$ for $1 \leq i < n-m$ as shown by Equation (2.3) Thus:

$$y_{1_{ss}} = \frac{b_0}{a_0} \tag{3.9}$$

$\underline{x}_0$ is now readily obtained from Equation (3.3) by setting t=0:

$$\underline{x}_0 = -\underline{y}_{ss} \tag{3.10}$$

since $\underline{y}(0) = 0$. Using Equations (3.7a) and (3.9) and substituting for $\underline{c}$ from Equation (2.3) gives the following result:

$$\begin{aligned} x_{1_0} &= -\frac{b_0}{a_0} \\ x_{(i+1)_0} &= 0 \qquad\qquad 1 \leq i < n-m \\ x_{(i+1)_0} &= b_{n-i} - \sum_{j=n-m}^{i-1} a_{n-i+j}\, x_{(j+1)_0} \qquad n-m \leq i < n \end{aligned} \tag{3.11}$$

where the summation term is zero when n-m>i-1. Since $\underline{x}_0$ is a function of the numerator coefficients of the transfer function given by Equation (3.1), it contains the effects of the system zeros as well as the input step function. The zeros are in fact represented by the last m initial condition states, which are referred to as pseudo initial conditions by Rediess [31] since they are not actual system initial conditions. The advantage of using the transient state vector is that it approaches zero as time increases for a stable system, which makes it a suitable variable in the integrand of a performance index defined as an integral over all time. The homogeneous equation is also simpler from a computational point of view than the equation, containing the forcing term explicitly.

It should be noted here that the transfer function given by Equation (3.1) is the closed-loop transfer function of the system.

The coefficients of its numerator and denominator are, therefore, known functions of the design parameters and will be written in vector form as $\underline{b}$ and $\underline{a}$, respectively. The design parameters, which are of most direct interest are the free design parameters, denoted by the vector $\underline{p}$, and the variable or uncertain parameters, denoted by $\underline{\xi}$. The free design parameters can be chosen by the designer to satisfy the system specifications. The variable parameters are assumed to have some known statistical distribution and may or may not be under some control of the designer. Thus the $\underline{\xi}$ vector may contain one of the elements of the $\underline{p}$ vector, for instance, in which case only the nominal value of that free design parameter can be chosen. A convenient method for dealing with this case will be developed later. In many cases, however, the variable parameters are completely beyond direct control of the designer.

The functional relationship between these design parameters and the closed loop system coefficients is then expressed by:

$$\underline{a} = \underline{a}\ (\underline{p},\underline{\xi})$$

and

$$\underline{b} = \underline{b}\ (\underline{p},\underline{\xi})$$

3.3 Problem Formulation

In the parameter optimization problem all feedback loops are closed beforehand, and the type of compensation selected. The system matrix in Equation (3.5) is, therefore, a function of the specified free design parameters, as well as the variable parameters, through its dependence on the coefficients of the characteristic equation,

$$\underline{A} = \underline{A}\ (\underline{p},\underline{\xi})$$

The initial conditions are also functions of these design parameters due to their dependence on the closed-loop system coefficients:

$$\underline{x}_0 = \underline{x}_0\ (\underline{p}, \underline{\xi})$$

The performance of the system is taken to be represented by a quadratic expression in terms of the transient system state vector:

$$J = \int_0^\infty \underline{x}(t)^T \underline{Q}\ \underline{x}(t)\ dt \tag{3.12}$$

where $\underline{Q}$ is a constant, positive semi-definite weighting matrix. Since the control input is contained in the homogeneous state equation, its effect is also included in this performance index. The optimum performance is achieved by minimizing J with respect to the free design parameters. When all system parameters are known to have some specified deterministic values, J is also deterministic and the minimization procedure is relatively straightforward. If, on the other hand, some system parameters are known only in terms of their statistical distribution, J is no longer deterministic. In this case it is logical to define the performance index to be the mathematical expectation of J:

$$\overline{J} = \int_0^\infty \overline{\underline{x}(t)^T \underline{Q}\ \underline{x}(t)}\ dt \tag{3.13}$$

where the bar represents the ensemble average of the quantities. It will be assumed that the variable parameters have some statistical distribution whose first and second order statistics are known. The nominal operating condition of the system will, furthermore, be chosen to correspond to the expected values of these parameters:

$$\underline{\xi}_* = \overline{\underline{\xi}}$$

where $\underline{\xi}$ is the vector of variable parameters and the asterisk indicates the nominal value. The deviation of $\underline{\xi}$ from its nominal value is then described by its covariance matrix:

$$\overline{\delta\underline{\xi}\ \delta\underline{\xi}^T} = \underline{R}$$

where $\delta\underline{\xi}$ is the deviation of $\underline{\xi}$ from the nominal value and $\underline{R}$ is a known matrix.

It should be mentioned again, that some of the components of $\underline{p}$ and $\underline{\xi}$ may be common, i.e. any free design parameter may also be a variable parameter. The nominal value of this parameter can then be chosen, but the variation about this value is determined by its variance. It is convenient in this case to write the true value of the free design parameter as:

$$p = p_* \xi$$

where $\underline{p}_*$ is the nominal value, which can be chosen, and ξ represents the variable part of this parameter, whose nominal value is given by:

$$\xi_* = 1$$

It is indeed, convenient for computational work to write all the variable parameters in this form, as a product of their nominal values and a variable part which then takes on the percentage deviation of the parameter. The system state vector can be written as:

$$\underline{x}(t) = \underline{x}_*(t) + \delta\underline{x}(t) \tag{3.14}$$

where $\underline{x}_*$ is the nominal response obtained from the system equation by using the nominal values of the variable parameters. The change in the response, $\delta\underline{x}$, is the result of some parameter variation, $\delta\underline{\xi}$, and is described to first order by the equation:

$$\delta\dot{\underline{x}} = \underline{A}_*\delta\underline{x} + \delta\underline{A}\,\underline{x}_* \quad ; \quad \delta\underline{x}(0) = \delta\underline{x}_0 \tag{3.15}$$

which is obtained by taking the first order variation of Equation (3.5). $\underline{A}_*$ is the nominal value of the system matrix and $\delta\underline{A}$ is its variation from the nominal, due to $\delta\underline{\xi}$, since $\underline{A}$ is a function of both $\underline{\xi}$ and the free design parameters. The initial state vector, $\underline{x}_0$, is also a function of $\underline{\xi}$ through its dependence on the coefficients of the transfer function. Its first order variation, $\delta\underline{x}_0$, can be expressed as:

$$\delta\underline{x}_0 = \left[\frac{\partial\underline{x}_0}{\partial\underline{\xi}}\right]_* \delta\underline{\xi} \tag{3.16}$$

where the derivative is evaluated for the nominal condition. By taking the expectation on both sides of Equation (3.15) and interchanging the order of that operation with the differentiation:

$$\overline{\delta\dot{\underline{x}}} = \underline{A}_*\,\overline{\delta\underline{x}} + \overline{\delta\underline{A}}\,\underline{x}_* \tag{3.17}$$

since $\underline{A}_*$ and $\underline{x}_*$ are both deterministic. Furthermore, by taking the variation and expected value of $\underline{A}$ in the phase-variable form:

$$\overline{\delta\underline{A}} = \left[\begin{array}{ccc} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \\ \hline & -\overline{\delta\underline{a}}^T & \end{array}\right] \tag{3.18}$$

But $\overline{\delta\underline{a}}$ can be written as:

$$\overline{\delta\underline{a}} = \left[\frac{\partial\underline{a}}{\partial\underline{\xi}}\right]_* \overline{\delta\underline{\xi}} = \underline{0} \Rightarrow \overline{\delta\underline{A}} = \underline{0} \tag{3.19}$$

since the derivative is deterministic and $\overline{\delta\underline{\xi}} = \underline{0}$ by the choice of the nominal parameter values. Similarly, by taking the expected value of Equation (3.16):

$$\overline{\delta \underline{x}_0} = \left[\frac{\partial \underline{x}_0}{\partial \underline{\xi}}\right]_* \overline{\delta \underline{\xi}} = \underline{0} \tag{3.20}$$

These results are substituted into Equation (3.17):

$$\overline{\delta \dot{\underline{x}}} = \underline{A}_* \overline{\delta \underline{x}} \ ; \quad \overline{\delta \underline{x}_0} = \underline{0} \tag{3.21}$$

This shows clearly that $\overline{\delta \underline{x}(t)} = \underline{0}$ since the equation has no forcing terms and is at rest initially. The expected value of the system response is, therefore, identical to the nominal response, $\underline{x}_*(t)$.

The performance index can now be expressed as:

$$J = \int_0^\infty (\underline{x}_*+\delta\underline{x})^T \underline{Q}(\underline{x}_*+\delta\underline{x})\, dt = \tag{3.22}$$

$$\int_0^\infty [\underline{x}_*^T \underline{Q}\ \underline{x}_*+\delta\underline{x}^T \underline{Q}\ \delta\underline{x} + 2\ \underline{x}_*^T \underline{Q}\ \delta\underline{x}]\ dt$$

Taking expected values on both sides of this equation:

$$\overline{J} = \int_0 [\underline{x}_*^T \underline{Q}\ \underline{x}_* + \overline{\delta\underline{x}^T \underline{Q}\ \delta\underline{x}}]\ dt \tag{3.23}$$

using the fact that $\overline{\delta \underline{x}} = \underline{0}$ and $\underline{Q}$ and $\underline{x}_*$ are deterministic. Hence, the expected value of the performance index is a sum of two terms, the first of which is its nominal value corresponding to the nominal system parameters. The second term represents the effect of the uncertainty of the parameter values on $\overline{J}$.

It is necessary at this point to address oneself to the problem of computing the value of $\overline{J}$, given the mean values of the variable parameters as well as their covariance matrix. For this purpose it is convenient to write $\overline{J}$ in the form:

$$\bar{J} = \mathrm{tr}\left[\underline{Q}\ \left(\int_0^\infty \underline{x}_*\underline{x}_*^T\ dt + \int_0^\infty \overline{\delta\underline{x}\ \delta\underline{x}^T}\ dt\right)\right]$$

$$= \mathrm{tr}\left[\underline{Q}\ (\underline{X} + \overline{\delta\underline{X}})\right] \tag{3.24}$$

where tr denotes the trace of the quantities and $\underline{X}$ and $\underline{\delta X}$ are defined as:

$$\underline{X} = \int_0^\infty \underline{x}_*\underline{x}_*^T\ dt$$

$$\delta\underline{X} = \int_0^\infty \delta\underline{x}\ \delta\underline{x}^T\ dt$$

The equations for $\underline{X}$ and $\delta\underline{X}$ must now be obtained. Using Equation (3.5) the following equation can be obtained for the integrand of $\underline{X}$:

$$\frac{d}{dt}(\underline{x}_*\underline{x}_*^T) = \underline{x}_*\dot{\underline{x}}_*^T + \dot{\underline{x}}_*\underline{x}_*^T = \underline{x}_*\underline{x}_*^T\ \underline{A}_*^T + \underline{A}_*\underline{x}_*\underline{x}_*^T \tag{3.25}$$

Integrating this equation on both sides from t=0 to infinity, gives:

$$\underline{x}_*\underline{x}_*^T(t=\infty) - \underline{x}_*\underline{x}_*^T\ (t=0) = \underline{X}\ \underline{A}_*^T + \underline{A}_*\underline{X} \tag{3.26}$$

using the definition of $\underline{X}$. Since $\underline{x}_*(t)$ is the solution of a linear homogeneous differential equation, its value approaches zero as $t \to \infty$ for a stable system. Thus, $\underline{x}_*\underline{x}_*^T(\infty) = \underline{0}$ and Equation (3.26) becomes:

$$\underline{A}_*\underline{X} + \underline{X}\ \underline{A}_*^T + \underline{X}_0 = \underline{0} \tag{3.27}$$

where

$$\underline{X}_0 = \underline{x}_*(0)\ \underline{x}_*^T(0)$$

$\underline{x}_*(0)$ is the initial condition of the nominal state vector and is, therefore, known. The equation for $\delta\underline{X}$ can be obtained in a similar manner. Using Equation (3.15) the following relationship is obtained:

$$\frac{d}{dt}(\delta\underline{x}\ \delta\underline{x}^T) = \delta\dot{\underline{x}}\ \delta\underline{x}^T + \delta\underline{x}\ \delta\dot{\underline{x}}^T$$

$$= \underline{A}_*\delta\underline{x}\ \delta\underline{x}^T + \delta\underline{A}\ \underline{x}_*\delta\underline{x}^T + \delta\underline{x}\delta\underline{x}^T\underline{A}_*^T + \delta\underline{x}\ \underline{x}_*^T\ \delta\underline{A}^T \qquad (3.28)$$

Integrating on both sides of this equation over all time gives:

$$\delta\underline{x}\delta\underline{x}^T(t=\infty) - \delta\underline{x}\delta\underline{x}^T(t=0) = \underline{A}_*\delta\underline{X} + \delta\underline{X}\ \underline{A}_*^T + \delta\underline{A}\underline{Y} + \underline{Y}^T\delta\underline{A}^T \qquad (3.29)$$

where $\underline{Y}$ is defined as:

$$\underline{Y} = \int_0^\infty \underline{x}_*\delta\underline{x}^T\ dt$$

Since $\lim_{t\to\infty} \underline{x}(t)=\underline{0}$ for a stable system, the variation of the state also goes to zero at infinity, $\lim_{t\to\infty} \delta\underline{x}(t)=\underline{0}$. Equation (3.29) then becomes:

$$\underline{A}_*\delta\underline{X} + \delta\underline{X}\ \underline{A}_*^T + \delta\underline{X}_0 + \delta\underline{A}\underline{Y} + \underline{Y}^T\delta\underline{A}^T = \underline{0} \qquad (3.30)$$

where $\delta\underline{X}_0$ is defined by:

$$\delta\underline{X}_0 = \delta\underline{x}(0)\delta\underline{x}^T(0) = \delta\underline{x}_0\delta\underline{x}_0^T$$

$\delta\underline{X}_0$ can be computed by using Equation (3.16), but $\underline{Y}$ remains to be determined. By using Equations (3.5) and (3.15) the following equation is obtained:

$$\frac{d}{dt}(\underline{x}_*\delta\underline{x}^T) = \dot{\underline{x}}_*\delta\underline{x}^T + \underline{x}_*\delta\dot{\underline{x}}^T = \underline{A}\ \underline{x}_*\delta\underline{x}^T + \underline{x}_*\delta\underline{x}^T\ \underline{A}_*^T + \underline{x}_*\underline{x}_*^T\delta\underline{A}^T \qquad (3.31)$$

which by integration over all time becomes:

$$\underline{x}_* \; \delta\underline{x}^T(t=\infty) \; - \; \underline{x}_* \; \delta\underline{x}(t=0) \; = \; \underline{A}_*\underline{Y} + \underline{Y}\,\underline{A}_*^T + \underline{X}\;\delta\underline{A}^T \tag{3.32}$$

Since $\lim_{t\to\infty} \underline{x}(t)=\underline{0}$, we finally obtain:

$$\underline{A}_*\underline{Y} + \underline{Y}\,\underline{A}_*^T + \underline{X}\;\delta\underline{A}^T + \underline{Y}_0 = \underline{0} \tag{3.33}$$

where $\underline{Y}_0$ is given by:

$$\underline{Y}_0 = \underline{x}_*(0)\;\delta\underline{x}^T(0)$$

With $\delta\underline{A}$ known and $\underline{X}$ obtained as a solution of Equation (3.27), $\underline{Y}$ can be found as the solution to this equation. This solution is then used in Equation (3.30) to solve for $\delta\underline{X}$. In order to find the mean value of J, however, it is necessary to determine the mean value of $\delta\underline{X}$ as seen from Equation (3.24). Taking the expected value of the terms in Equation (3.30) gives:

$$\underline{A}_*\;\overline{\delta\underline{X}} + \overline{\delta\underline{X}}\;\underline{A}_*^T + \overline{\delta\underline{X}_0} + \overline{\delta\underline{A}\underline{Y}} + \overline{\underline{Y}^T\;\delta\underline{A}^T} = \underline{0} \tag{3.34}$$

$\delta\overline{\underline{X}}_0$ can be obtained by using its definition and Equation (3.16):

$$\delta\overline{\underline{X}}_0 = \left[\frac{\partial\underline{x}_0}{\partial\underline{\xi}}\right]_* \overline{\delta\underline{\xi}\;\delta\underline{\xi}^T} \left[\frac{\partial\underline{x}_0}{\partial\underline{\xi}}\right]_* = \left[\frac{\partial\underline{x}_0}{\partial\underline{\xi}}\right]_* \underline{R} \left[\frac{\partial\underline{x}_0}{\partial\underline{\xi}}\right]_*^T \tag{3.35}$$

The quantity $\overline{\delta \underline{A}\underline{Y}}$ and its transpose present a problem, however, since $\delta\underline{A}$ and $\underline{Y}$ are both functions of the variable parameters, which results in their being correlated. One possible method for obtaining the expected value of this matrix product is to premultiply Equation (3.33) by $\delta\underline{A}$ and take the expected value of each term:

$$\overline{\delta\underline{A}\,\underline{A}_*\underline{Y}} + \overline{\delta\underline{A}\;\underline{Y}}\;\underline{A}_*^T + \overline{\delta\underline{A}\;\underline{X}\;\delta\underline{A}^T} + \overline{\delta\underline{A}\underline{Y}_0} = \underline{0} \tag{3.36}$$

The last two terms on the left hand side can be computed from the parameter covariance matrix, since $\underline{X}$ is a deterministic matrix. The equation can then, in theory, be solved in terms of the elementary products which make up the components of the product $\overline{\delta\underline{A}\underline{Y}}$. This would be a very difficult task at best and a much simpler method can be developed by taking advantage of the form of the system matrix and its variation. The product of $\delta\underline{A}$ and $\underline{Y}$ can then be written as:

$$\delta\underline{A}\underline{Y} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \\ \hline & -\delta\underline{a}^T & \end{bmatrix} \begin{bmatrix} | & | & & | \\ \underline{y}_0 & \underline{y}_1 & \cdots & \underline{y}_{n-1} \\ | & | & & | \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \\ -\delta\underline{a}^T\underline{y}_0 & \cdots & -\delta\underline{a}^T\underline{y}_{n-1} \end{bmatrix} \tag{3.37}$$

where $\underline{Y}$ has been written in terms of its column vectors. Thus, it is only necessary to find the values of $\overline{\delta\underline{a}^T y_i}$ for $0 \leq i \leq n-1$ in order to completely determine $\overline{\delta\underline{A}\underline{Y}}$. This can be done by expanding Equation (3.33), considering each of its columns separately, which results in n vector equations:

$$\begin{aligned} &\underline{A}_*\,\underline{y}_0 + \underline{y}_1 + \delta x_{1_0}\underline{x}_0^* = \underline{0} \\ &\vdots \\ &\underline{A}_*\underline{y}_{n-2} + \underline{y}_{n-1} + \delta x_{(n-1)_0}\underline{x}_0^* = \underline{0} \\ &\underline{A}_*\underline{y}_{n-1} - a_{n-1}^*\underline{y}_{n-1} - \cdots - a_0^*\underline{y}_0 + \delta x_{n_0}\underline{x}_0^* - \underline{x}\delta\underline{a} = \underline{0} \end{aligned} \tag{3.38}$$

The first (n-1) equations are iterative and make it possible to determine the $i^{\underline{th}}$ column vector of $\underline{Y}$ in terms of its $(i-1)^{\underline{st}}$ column vector.

Premultiplying these equations by $\delta\underline{a}^T$ and taking the expected value yields n scalar equations which are not sufficient in order to solve for the n desired scalar products $\overline{\delta\underline{a}^T\underline{y}_i}$, since these equations contain other product terms of the form $\overline{\delta a_i y_j}$. One can, however, express the inner product of $\delta\underline{a}$ and $\underline{y}_i$ as the trace of their outer product:

$$\overline{\delta\underline{a}^T\underline{y}_i} = \text{tr}\left[\overline{\underline{y}_i\delta\underline{a}^T}\right] \tag{3.39}$$

The following iterative relationship is then obtained by postmultiplying the first (n-1) equations of (3.38) by $\delta\underline{a}^T$ and taking expected values:

$$\underline{Z}_i + \underline{A}_*\underline{Z}_{i-1} + \underline{x}_0^*\underline{v}_{i-1}^T = \underline{0} \qquad 0<i\leq n-1 \tag{3.40}$$

where

$$\underline{Z}_i = \overline{\underline{y}_i \ \delta\underline{a}^T} \tag{3.41}$$

and $\underline{v}_i$ is the $(i+1)^{\underline{st}}$ column vector of the matrix $\underline{V}$, which is defined as:

$$\underline{V} = \overline{\delta\underline{a} \ \delta\underline{x}_0^T} = \left[\frac{\partial\underline{a}}{\partial\underline{\xi}}\right]_* \underline{R} \left[\frac{\partial\underline{x}_0}{\partial\underline{\xi}}\right]_*^T \tag{3.42}$$

The last equation of (3.38) becomes, in a similar way:

$$\underline{A}_*\underline{Z}_{n-1} - a_{n-1}^*\underline{Z}_{n-1} - \ldots - a_0^*\underline{Z}_0 + \underline{x}_0^*\underline{v}_{n-1}^T - \underline{X}\,\underline{W} = \underline{0} \tag{3.43}$$

where

$$\underline{W} = \overline{\delta\underline{a}\;\delta\underline{a}^T} = \left[\frac{\partial \underline{a}}{\partial \underline{\xi}}\right]_* \underline{R} \left[\frac{\partial \underline{a}}{\partial \underline{\xi}}\right]_*^T \tag{3.44}$$

By solving Equations (3.40) and (3.43) it is, therefore, possible to determine all the terms necessary for solving Equation (3.34), which can now be written as follows:

$$\underline{A}_* \overline{\delta\underline{X}} + \overline{\delta\underline{X}}\,\underline{A}_*^T = - [\underline{U} + \overline{\delta\underline{X}_0}] \tag{3.45}$$

where

$$\underline{U} = - \begin{bmatrix} 0 & \cdots & 0 & tr\underline{Z}_0 \\ \vdots & & \vdots & tr\underline{Z}_1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & tr\underline{Z}_{n-2} \\ tr\underline{Z}_0 & tr\underline{Z}_1 \cdots & tr\underline{Z}_{n-2} & 2tr\underline{Z}_{n-1} \end{bmatrix} \tag{3.46}$$

The value of the performance index, $\overline{J}$, can therefore be computed by solving a series of algebraic matrix equations. The task of finding the minimum value of $\overline{J}$ with respect to the design parameters is discussed in detail in the next chapter.

3.3.1 Discussion of Performance Index

The approach taken in this section is basically that of the stochastic control problem where the uncertainty is due to the statistical nature of some system parameters instead of random noise, which enters the system as an input. Under the assumption of first order variations the performance index separates into two terms as seen from Equation (3.23). The first term is the value of the performance index when the system parameters are deterministic and take on their nominal values. The second term represents the effect of the parameter variations on the expected value of the performance index. It is of the same form as the sensitivity index defined by Equation (2.10) in the previous chapter. Using the expected value of the performance index thus leads to a similar expression as given by Equation (2.35) where a sensitivity index is added to the quadratic performance index representing the nominal system performance. The weighting matrix of the sensitivity index has, furthermore, been determined as being equal to the weighting matrix of the nominal state. This does not preclude the possibility of using a different weighting matrix for the sensitivity index, since in some applications it may be desirable to change the relative importance of the various state deviations. In addition it could also be of interest to change the relative weighting of the sensitivity term with respect to the nominal term. The performance index may then be written as:

$$\bar{J} = \int_0^\infty \underline{x}_*^T Q_1 \underline{x}_* dt + \varepsilon \int_0^\infty \overline{\delta \underline{x}^T \underline{Q}_2 \delta \underline{x}}\, dt \tag{3.47}$$

where $\underline{Q}_1$ and $\underline{Q}_2$ are not necessarily equal and ε is an arbitrary weighting constant. Changing the value of ε is, however, completely equivalent to scaling the covariance matrix of the variable parameters in the same proportion, since:

$$\overline{\delta \underline{x} \delta \underline{x}^T} = \left[\frac{\partial \underline{x}}{\partial \underline{\xi}}\right]_* \overline{\delta \underline{\xi} \delta \underline{\xi}^T} \left[\frac{\partial \underline{x}}{\partial \underline{\xi}}\right]_*^T \tag{3.48}$$

where the derivatives are evaluated on the nominal response trajectory. So putting more emphasis on the sensitivity index is equivalent to increasing the spread of the joint distribution of the variable parameters.

The approach taken to the sensitivity problem in this thesis ties together the stochastic control approach and the technique of adding an index of sensitivity to the performance index used to express the nominal performance. In the past, both of these methods have suffered from the difficulties associated with obtaining a numerical solution to practical problems, especially when more than one system parameter is involved. The formulation given here alleviates these difficulties to a great extent as will be seen in the following chapter. This is largely possible due to the convenient form of the equations when the state equation is written in the phase-variable form.

3.4 Necessary Conditions

In the previous section the following equations were derived for determining the performance index as defined by the quadratic form in the system state:

$$
\begin{aligned}
&\bar{J} = \mathrm{tr}\left[\underline{Q}[\underline{X} + \delta\underline{\bar{X}}]\right] \\
&\underline{A}_*\underline{X} + \underline{X}\underline{A}_*^T + \underline{X}_0 = \underline{0} \\
&\underline{A}_*\delta\underline{\bar{X}} + \delta\underline{\bar{X}}\underline{A}_*^T + \underline{U} + \delta\underline{\bar{X}}_0 = \underline{0} \qquad\qquad (3.49) \\
&\underline{Z}_i + \underline{A}_*\underline{Z}_{i-1} + \underline{x}_0^* \underline{v}_{i-1}^T = \underline{0} \qquad\qquad 0<i\underline{<}n-1 \\
&\underline{A}_*\underline{Z}_{n-1} - a_{n-1}^*\underline{Z}_{n-1} - \ldots - a_0^*\underline{Z}_0 + \underline{x}_0^*\underline{v}_{n-1}^T - \underline{X}\,\underline{W} = 0
\end{aligned}
$$

where all system coefficients are known as functions of the variable parameters as well as the free design parameters.†

---

†From here on all system coefficients, as well as functions and derivatives of these coefficients, will be evaluated using nominal values of the variable parameters. The (*) notation will therefore be dropped.

The necessary condition for $\overline{J}$ to be at a local minimum for specified values of the design parameters is that its variation with respect to these parameters be zero to first order. A standard method for obtaining the variation of a functional in the presence of contraining equations is to add the constraints to the functional by the use of Lagrange multipliers. Thus the performance index is augmented to give:

$$\overline{J} = \mathrm{tr}\Big\{\underline{Q}\underline{X} + \underline{Q}\delta\overline{\underline{X}} + \underline{P}_1[\underline{A}\underline{X} + \underline{X}\underline{A}^T + \underline{X}_0] + \underline{P}_2[\underline{A}\delta\overline{\underline{X}} + \delta\overline{\underline{X}}\underline{A}^T + U + \delta\overline{\underline{X}}_0]$$

$$+ \sum_{i=1}^{n-1} \underline{\Lambda}_i \; [\underline{Z}_i + \underline{A}\underline{Z}_{i-1} + \underline{x}_0\underline{v}_{i-1}^T] + \underline{\Lambda}_n \; [\underline{A}\underline{Z}_{n-1} - a_{n-1}\underline{Z}_{n-1} - \ldots$$

$$-a_0\underline{Z}_0 + \underline{x}_0\underline{v}_{n-1}^T - \underline{X}\underline{W}]\Big\} \tag{3.50}$$

where $\underline{P}_1$, $\underline{P}_2$, and $\underline{\Lambda}_i$ are n×n matrices of Lagrange multipliers. The traces of the matrix products, which are added to the performance index, are sufficient to constrain each element of the matrix equations. This is seen from the fact that the trace of a matrix product can be written as:

$$\mathrm{tr}\;[\underline{P}\underline{M}] = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}\, m_{ji}$$

where the right hand side consists of the sum of simple products in terms of the elements of $\underline{P}$ and $\underline{M}$ with no common factors in any two product terms. This is just the type of expression needed to constrain each element of $\underline{M}$ to zero.

The variation with respect to the free design parameters will be denoted by $\tilde{\delta}$ in order to distinguish it from the variation due to the variable system parameters, which is indicated by δ. The necessary conditions are determined by requiring the variation of $\overline{J}$ with respect to each of the quantities, $\underline{X}$, $\delta\overline{\underline{X}}$ and $\underline{Z}_i$, to be equal to zero. Thus, considering the effect on $\bar{J}$ due to a variation of $\underline{X}$ we get:

$$\tilde{\delta}\bar{J} = \mathrm{tr}\left[[\underline{Q} + (\underline{P}_1\underline{A} + \underline{A}^T\underline{P}_1) - \underline{W}\underline{\Lambda}_n]\ \tilde{\delta}\underline{X} \right] = 0 \tag{3.51}$$

where the following matrix property has been used.

$$\mathrm{tr}\ [\underline{A}\underline{B}] = \mathrm{tr}[\underline{B}\underline{A}]$$

where $\underline{AB}$ is a square matrix. The following equation is then obtained for $\underline{P}_1$, since $\tilde{\delta}\underline{X}$ is, in general, non-zero:

$$\underline{P}_1\underline{A} + \underline{A}^T\underline{P}_1 = \underline{W}\underline{\Lambda}_n - \underline{Q} \tag{3.52}$$

The variation with respect to $\delta\underline{\bar{X}}$ is, similarly:

$$\tilde{\delta}\underline{\bar{J}} = \mathrm{tr}\left[[\underline{Q} + (\underline{P}_2\underline{A} + \underline{A}^T\underline{P}_2)]\ \tilde{\delta}(\delta\underline{\bar{X}}) \right] = 0 \tag{3.53}$$

which results in the equation for $\underline{P}_2$:

$$\underline{P}_2\underline{A} + \underline{A}^T\underline{P}_2 = -\underline{Q} \tag{3.54}$$

In order to find the variations due to the $\underline{Z}_i$ matrices it is necessary to rewrite the term $\mathrm{tr}(\underline{P}_2\underline{U})$ in Equation (3.50) since $\underline{U}$ is a function of the $\underline{Z}_i$ matrices as expressed by Equation (3.46). $\underline{P}_2$ is a symmetric matrix as can be seen by transposing Equation (3.60) and using the fact that $\underline{Q}$ is symmetric. The product of $\underline{P}_2$ and $\underline{U}$ can then be written:

$$\underline{P}_2\underline{U} = - \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{12} & p_{22} & \cdots & p_{2n} \\ \vdots & & & \vdots \\ p_{1n} & p_{2n} & \cdots & p_{nn} \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 & trZ_0 \\ \vdots & & \vdots & trZ_1 \\ 0 & \cdots & 0 & \vdots \\ trZ_0 & trZ_1 & \cdots & 2trZ_{n-1} \end{bmatrix} \qquad (3.55)$$

The trace of this product can then be expressed as:

$$tr(\underline{P}_2\underline{U}) = -2 \sum_{i=1}^{n} p_{in} \, trZ_{i-1} = -2 \sum_{i=1}^{n} tr(p_{in}Z_{i-1}) \qquad (3.56)$$

The variation of $\bar{J}$ with respect to $\underline{Z}_0$ is then:

$$\tilde{\delta}\bar{J} = tr\left[[-2p_{1n}\underline{I} + \underline{\Lambda}_1\underline{A} - a_0\underline{\Lambda}_n] \; \tilde{\delta}\underline{Z}_0\right] = 0 \qquad (3.57)$$

which results in the following relationship between $\underline{\Lambda}_1$ and $\underline{\Lambda}_n$:

$$\underline{\Lambda}_1\underline{A} - a_0\underline{\Lambda}_n - 2p_{1n}\underline{I} = \underline{0} \qquad (3.58)$$

Taking the variation with respect to Z_{i-1} for $1<i\leq n$ gives:

$$\tilde{\delta}\bar{J} = tr\left[[-2p_{in}\underline{I} + \underline{\Lambda}_i\underline{A} + \underline{\Lambda}_{i-1} - a_{i-1}\underline{\Lambda}_n] \; \tilde{\delta}\underline{Z}_{i-1}\right] = 0 \qquad (3.59)$$

which then leads to the equation:

$$\underline{\Lambda}_{i-1} + \underline{\Lambda}_i\underline{A} - a_{i-1}\underline{\Lambda}_n - 2p_{in}\underline{I} = \underline{0} \qquad 1<i\leq n \qquad (3.60)$$

Equations (3.58) and (3.60) can now be used to solve for all the $\underline{\Lambda}_i$ matrices.

Finally it remains to determine the variation of $\bar{J}$ due to all the terms which are explicit functions of the free design parameters. These are all the terms containing the system coefficients and initial conditions or perturbations of these with respect to the variable system parameters. Thus:

$$\tilde{\delta}\bar{J} = \mathrm{tr}\Big[(\underline{P}_1 + \underline{P}_1^T)\ \underline{X}\tilde{\delta}\underline{A}^T + \underline{P}_1\tilde{\delta}\underline{X}_0 + (\underline{P}_2 + \underline{P}_2^T)\overline{\delta\underline{X}}\ \tilde{\delta}\underline{A}^T$$

$$+ \underline{P}_2\tilde{\delta}(\overline{\delta\underline{X}}_0) + [\sum_{i=1}^{n} \underline{\Lambda}_i^T\underline{Z}_{i-1}^T]\ \tilde{\delta}\underline{A}^T + \sum_{i=1}^{n} \underline{\Lambda}_i(\tilde{\delta}\underline{x}_0\underline{v}_{i-1}^T + \underline{x}_0\tilde{\delta}\underline{v}_{i-1}^T)$$

$$- \sum_{i=1}^{n} [\underline{\Lambda}_n\underline{Z}_{i-1}\ \tilde{\delta}a_{i-1}] - \underline{\Lambda}_n\underline{X}\tilde{\delta}\underline{W}\Big] = 0 \qquad (3.61)$$

where some of the terms have been rearranged using the matrix identity:

$$\mathrm{tr}(\underline{A}\underline{B}) = \mathrm{tr}(\underline{B}^T\underline{A}^T) = \mathrm{tr}(\underline{A}^T\underline{B}^T)$$

It is necessary to express the variations of the quantities in Equation (3.61) in terms of the first order variation of the free design parameters, $\tilde{\delta}\underline{p}$. Using the definition of $\underline{A}$, the following expression is obtained:

$$\tilde{\delta}\underline{A}^T = -\tilde{\delta}\underline{a}\underline{\eta}^T = -\frac{\partial \underline{a}}{\partial \underline{p}}\ \tilde{\delta}\underline{p}\ \underline{\eta}^T \qquad (3.62)$$

where $\underline{\eta}^T$ is an n-dimensional vector defined by:

$$\underline{\eta}^T = [\ 0, 0, \ldots, 0, 1\ ]$$

Thus, all the terms in $\tilde{\delta}\bar{J}$ containing $\tilde{\delta}\underline{A}$ can be rearranged in the following manner:

$$\mathrm{tr}\left[\underline{M}\tilde{\delta}\underline{A}^T\right] = -\underline{n}^T\underline{M} \left[\frac{\partial \underline{a}}{\partial \underline{p}}\right]\tilde{\delta}\underline{p} \tag{3.63}$$

Furthermore, using the definition of $\overline{\delta \underline{X}}_0$ as given in Section 3.3:

$$\tilde{\delta}\underline{X}_0 = \left[\frac{\partial \underline{x}_0}{\partial \underline{p}}\right] \tilde{\delta}\underline{p}\ \underline{x}_0^T + \underline{x}_0\tilde{\delta}\underline{p}^T \left[\frac{\partial \underline{x}_0}{\partial \underline{p}}\right]^T \tag{3.64}$$

The corresponding term in $\tilde{\delta}\overline{J}$ can then be rearranged:

$$\mathrm{tr}\left[\underline{P}_1\tilde{\delta}\underline{X}_0\right] = \underline{x}_0^T\ [\underline{P}_1 + \underline{P}_1^T] \left[\frac{\partial \underline{x}_0}{\partial \underline{p}}\right] \tilde{\delta}\underline{p} \tag{3.65}$$

The following equations are obtained in a similar fashion:

$$\mathrm{tr}\left[\underline{\Lambda}_i\ \tilde{\delta}\underline{x}_0\ \underline{v}_{i-1}^T\right] = \underline{v}_{i-1}^T\ \underline{\Lambda}_i \left[\frac{\partial \underline{x}_0}{\partial \underline{p}}\right] \tilde{\delta}\underline{p} \tag{3.66}$$

$$\mathrm{tr}\left[\underline{\Lambda}_i\underline{x}_0\tilde{\delta}\underline{v}_{i-1}^T\right] = \underline{x}_0^T\ \underline{\Lambda}_i^T\ \tilde{\delta}\underline{v}_{i-1} = \underline{x}_0^T\underline{\Lambda}_i^T \left[\frac{\partial \underline{v}_{i-1}}{\partial \underline{p}}\right] \tilde{\delta}\underline{p} \tag{3.67}$$

$$\mathrm{tr}\left[\underline{\Lambda}_n\underline{Z}_{i-1}\tilde{\delta}a_{i-1}\right] = \mathrm{tr}\left[\underline{\Lambda}_n\underline{Z}_{i-1}\right] \left[\frac{\partial a_{i-1}}{\partial \underline{p}}\right] \tilde{\delta}\underline{p} \tag{3.68}$$

Using the definition of $\underline{v}_{i-1}$, its derivative with respect to the j$\underline{\text{th}}$ parameter can be written:

$$\frac{\partial \underline{v}_{i-1}}{\partial p_j} = \left[\frac{\partial x_{i_0}}{\partial p_j \partial \underline{\xi}}\right] \underline{R} \left[\frac{\partial \underline{a}}{\partial \underline{\xi}}\right]^T + \left[\frac{\partial x_{i_0}}{\partial \underline{\xi}}\right] \underline{R} \left[\frac{\partial^2 \underline{a}}{\partial p_j \partial \underline{\xi}}\right]^T \tag{3.69}$$

which is the j$\underline{\text{th}}$ column of the derivative matrix in Equation (3.67). The terms containing $\overline{\delta \underline{X}}_0$ and $\underline{W}$ are somewhat more complicated. Using the expression for $\overline{\delta \underline{X}}_0$ given by Equation (3.64):

$$\mathrm{tr}\left[\underline{P}_2\tilde{\delta}(\overline{\delta \underline{X}}_0)\right] = \mathrm{tr}\left[\underline{P}_2 \left[\left[\frac{\partial \overline{\delta \underline{X}}_0}{\partial p_1}\right]\tilde{\delta}p_1, +\ldots+ \left[\frac{\partial \overline{\delta \underline{X}}_0}{\partial p_k}\right]\tilde{\delta}p_k\right]\right] \tag{3.69}$$

The variation due to the $j^{\underline{th}}$ parameter can then be written:

$$\mathrm{tr}\left[\underline{P}_2\left[\frac{\partial \delta\overline{\underline{X}}_0}{\partial p_j}\right]\delta p_j\right] = \mathrm{tr}\left[2\ \underline{P}_2\left[\frac{\partial^2 \underline{x}_0}{\partial p_j \partial \underline{\xi}}\right]\underline{R}\left[\frac{\partial \underline{x}_0}{\partial \underline{\xi}}\right]^T\right]\delta p_j \tag{3.70}$$

where the fact that $\underline{P}_2$ is symmetric has been used.

The total variation of this term can then be written:

$$\mathrm{tr}\left[\underline{P}_2\ \tilde{\delta}(\overline{\delta \underline{X}}_0)\right] = \underline{e}^T\ \tilde{\delta}\underline{p} \tag{3.71}$$

where

$$e_i = 2\ \mathrm{tr}\left[\underline{P}_2\left[\frac{\partial^2 \underline{x}_0}{\partial p_i \partial \underline{\xi}}\right]\underline{R}\left[\frac{\partial \underline{x}_0}{\partial \underline{\xi}}\right]^T\right]$$

Similarly we have that:

$$\mathrm{tr}\left[\underline{\Lambda}_n\underline{X}\ \frac{\partial \underline{W}}{\partial p_i}\delta p_i\right] = \mathrm{tr}\left[\left[\underline{\Lambda}_n\underline{X} + \underline{X}\underline{\Lambda}_n^T\right]\left[\frac{\partial^2 \underline{a}}{\partial p_i \partial \underline{\xi}}\right]\underline{R}\left[\frac{\partial \underline{a}}{\partial \underline{\xi}}\right]^T\right]\tilde{\delta}p_i \tag{3.72}$$

and

$$\mathrm{tr}\left[\underline{\Lambda}_n^T\ \underline{X}\ \tilde{\delta}\underline{W}\right] = \underline{f}^T\ \tilde{\delta}\underline{p} \tag{3.73}$$

where

$$f_i = \mathrm{tr}\left[\left[\underline{\Lambda}_n\underline{X} + \underline{X}\underline{\Lambda}_n^T\right]\left[\frac{\partial^2 \underline{a}}{\partial p_i \partial \underline{\xi}}\right]\underline{R}\left[\frac{\partial \underline{a}}{\partial \underline{\xi}}\right]^T\right]$$

Collecting all the above terms, the variation of $\overline{J}$ can be now be written in the form:

$$\tilde{\delta J} = \Bigg[-\underline{n}^T \Big[(\underline{P}_1+\underline{P}_1^T)\ \underline{X} + 2\underline{P}_2\ \overline{\delta \underline{X}} + \sum_{i=1}^{n} \underline{\Lambda}_i^T \underline{Z}_{i-1}^T \Big] \Big[\frac{\partial \underline{a}}{\partial \underline{p}} \Big]$$

$$+ \underline{x}_o^T\ [\underline{P}_1+\underline{P}_1^T] \Big[\frac{\partial \underline{a}}{\partial \underline{p}} \Big] + \underline{x}_0^T \sum_{i=1}^{n} \underline{\Lambda}_i^T \Big[\frac{\partial \underline{v}_{i-1}}{\partial \underline{p}} \Big]$$

$$+ \Big[\sum_{i=1}^{n} \underline{v}_{i-1}^T\ \underline{\Lambda}_i \Big] \Big[\frac{\partial \underline{x}_0}{\partial \underline{p}} \Big] - \sum_{i=1}^{n} \Big[\mathrm{tr}\,[\underline{\Lambda}_n \underline{Z}_{i-1}] \Big] \Big[\frac{\partial a_{i-1}}{\partial \underline{p}} \Big] + \underline{e}^T - \underline{f}^T \Bigg] \tilde{\delta p}$$

$$= 0 \qquad (3.74)$$

The expression in the brackets must be equal to zero for this equation to hold. All the necessary conditions for a local minimum of $\overline{J}$ are summarized below:

$$\underline{A}\underline{X} + \underline{X}\underline{A}^T + \underline{X}_0 = \underline{0}$$

$$\underline{A}\overline{\delta \underline{X}} + \overline{\delta \underline{X}}\ \underline{A}^T + \underline{U} + \overline{\delta \underline{X}}_0 = \underline{0}$$

$$\underline{Z}_i + \underline{A}\underline{Z}_{i-1} + \underline{x}_0 \underline{v}_{i-1}^T = \underline{0} \qquad 0<i\leq n-1$$

$$\underline{A}\underline{Z}_{n-1} - \sum_{i=0}^{n-1} a_i \underline{Z}_i + \underline{x}_0 \underline{v}_{n-1}^T - \underline{X}\underline{W} = \underline{0}$$

$$\underline{P}_1\underline{A} + \underline{A}^T\underline{P}_1 - \underline{W}\underline{\Lambda}_n + \underline{Q} = \underline{0}$$

$$\underline{P}_2\underline{A} + \underline{A}^T\underline{P}_2 + \underline{Q} = \underline{0}$$

$$\underline{\Lambda}_{i-1} + \underline{\Lambda}_i\underline{A} - a_{i-1}\underline{\Lambda}_n - 2p_{in}\underline{I} = \underline{0} \qquad 1<i\leq n$$

$$\underline{\Lambda}_1\underline{A} - a_0\underline{\Lambda}_n - 2p_{1n}\underline{I} = \underline{0}$$

$$\underline{\eta}^T \left[(\underline{P}_1+\underline{P}_1^T)\ \underline{X} + 2\underline{P}_2\overline{\delta \underline{X}} + \sum_{i-1}^{n} \underline{\Lambda}_i^T\underline{Z}_{i-1}^T\right]\left[\frac{\partial \underline{a}}{\partial \underline{p}}\right]$$

$$- \underline{x}_0^T\ [\underline{P}_1+\underline{P}_1^T] \left[\frac{\partial \underline{a}}{\partial \underline{p}}\right] - \underline{x}_0^T \sum_{i=1}^{n} \underline{\Lambda}_i^T \left[\frac{\partial \underline{v}_{i-1}}{\partial \underline{p}}\right]$$

$$- \left[\sum_{i=1}^{n} \underline{v}_{i-1}^T\underline{\Lambda}_i\right]\left[\frac{\partial \underline{x}_0}{\partial \underline{p}}\right] + \sum_{i-1}^{n} \mathrm{tr}[\underline{\Lambda}_n\underline{Z}_{i-1}]\left[\frac{\partial a_{i-1}}{\partial \underline{p}}\right] - \underline{e}^T + \underline{f}^T = \underline{0} \quad (3.75)$$

The set of free design parameters which satisfies all these equations determines a local minimum of the performance index $\overline{J}$. The method for numerically obtaining this solution is discussed in Chapter 4.

The next two sections are devoted to the discussion of two specific performance indices which have been found useful in system design. The first of these has the same form as the performance index which has been used in the previous sections, with a systematic procedure for determining the weighting matrix. The second performance index is the well-known integral square error criterion (ISE), which is defined in terms of the error response of the system as compared with a reference model response. The ISE thus contains the model response explicitly and this requires that the necessary conditions be augmented.

3.5 The Model Performance Index

The model performance index was formulated by Rediess [31] and gives a systematic method for determining the state weighting matrix in the quadratic performance index commonly used for optimizing the design of linear feedback control systems. When the performance index is minimized, the system response becomes close to or identical to that of a specified model response. Moreover, the model's time response is not included explicitly in the cost function. The unit step function is used as a standard input, although other inputs could be used. The original derivation of the model performance index given in Reference [31] is based on a geometrical interpretation of the system state equation in the phase-variable form. A somewhat different interpretation will be given in this section, based on the error state equations of the system where the error state is defined as the difference between the system state and the state of a reference model.

3.5.1 System Error Equation

The system transient response is described by the equations derived in Section 3.2:

$$\dot{\underline{x}} = \underline{A}\underline{x}; \quad \underline{x}(0) = \underline{x}_o \tag{3.5}$$

where the system matrix is in the phase-variable form and the initial condition vector contains the effects of the step response and system zeros. The desired system response is taken to be described by a model state equation of the same form:

$$\dot{\hat{\underline{x}}} = \hat{\underline{A}}\hat{\underline{x}} \quad ; \quad \hat{\underline{x}}(0) = \hat{\underline{x}}_0 \tag{3.76}$$

where (^) refers to the model. Assuming that the model is of the same order as the system, the error equation can be written:

$$\Delta\dot{\underline{x}} = \hat{\underline{A}}\,\Delta\underline{x} + \Delta\underline{A}\underline{x} \quad ; \Delta\underline{x}(0) = \Delta\underline{x}_0 \tag{3.77}$$

where

$$\Delta\underline{x} = \underline{x} - \hat{\underline{x}} \quad ; \Delta\underline{x}_0 = \underline{x}_0 - \hat{\underline{x}}_0$$

and

$$\Delta\underline{A} = \underline{A} - \hat{\underline{A}}$$

The homogeneous part of the equation is identical to the model equation with the model state replaced by the error state. The forcing term is expressed only in terms of the system state. The error response can then be obtained as shown in Figure 3.1.

In a practical situation, however, the model is commonly of a lower order than the system order. The same error equation can be used in this case but the model equation must be augmented such as to be compatible with the system equation. This can be done by observing the fact that the model's state space is a subspace of the system state space. The model equation in n-dimensional space can be written as:

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \\ \vdots \\ \dot{\hat{x}}_\ell \\ \dot{\hat{x}}_{\ell+1} \\ \vdots \\ \dot{\hat{x}}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & & \vdots & \vdots & & \vdots \\ -\alpha_0 & -\alpha_1 & & \cdots & -\alpha_{\ell-1} & 0 & \cdots & 0 \\ 0 & & \cdots & & 0 & & \cdots & 0 \\ \vdots & & & & \vdots & & & \vdots \\ 0 & & \cdots & & 0 & & \cdots & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \\ \vdots \\ \hat{x}_\ell \\ \hat{x}_{\ell+1} \\ \vdots \\ \hat{x}_n \end{bmatrix} ; \quad \hat{\underline{x}}_0 = \begin{bmatrix} \hat{x}_{1_0} \\ \vdots \\ \hat{x}_{\ell_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.78}$$

where ℓ is the order of the model ($\ell<n$) and the α's denote its characteristic coefficients. Thus, the dimension of the model equation has been made equal to the system order by the addition of zeros to the model matrix. The state vector has also been augmented to $n^{\underline{th}}$ dimension but it may be noted that the last $(n-\ell)$ states are identically zero for all time.

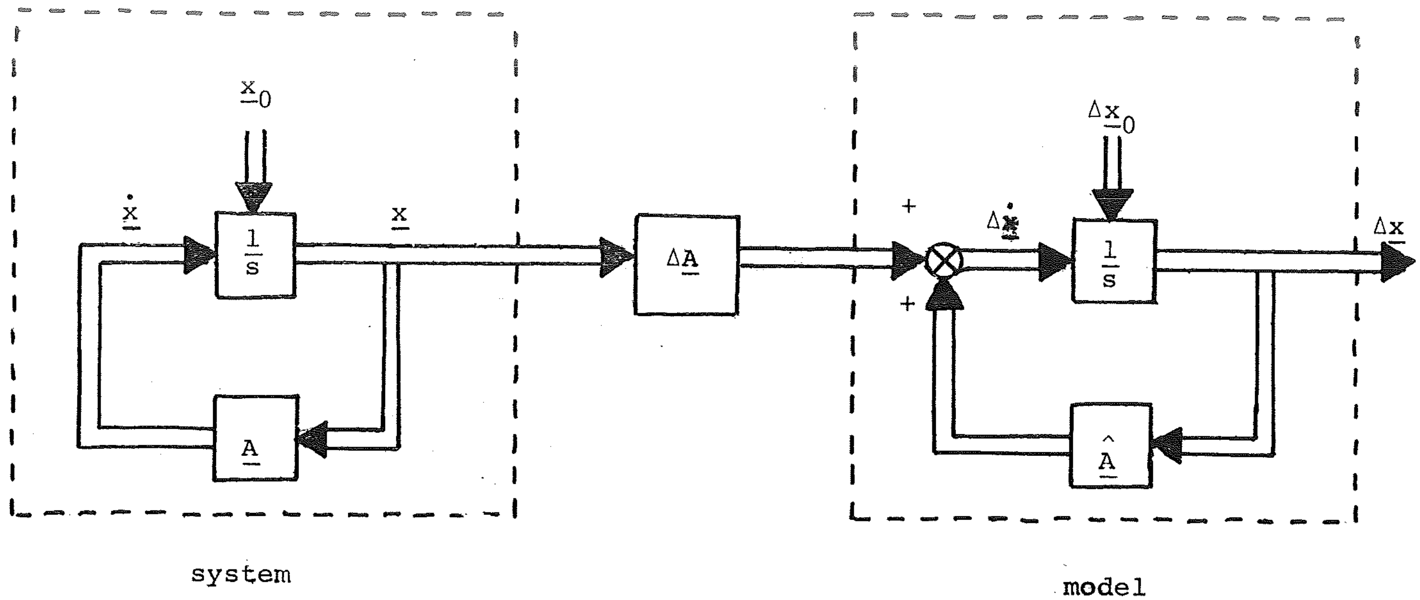


Figure 3.1. System Error Response

This augmented model equation can then be used in the error equation (3.77). It is clear that the last $(n-\ell)$ error states are equal to the corresponding system state variables:

$$\Delta x_i = x_i \qquad \ell < i \leq n$$

since the last $(n-\ell)$ model states are zero.

The error state equation will now be used to give a simple interpretation of the model performance index. First, only systems and models without zeros in their transfer function will be considered.

3.5.2 Systems without Zeros

It can be seen from the expression for the initial conditions Equation (3.11), that for systems which have no zeros in the transfer function, all but the first initial state are zero, since in this case m=0. This state is, furthermore, equal to the negative of the system's static sensitivity. Assuming that the system and model have equal static sensitivities, this initial error state is also zero. This assumption is reasonable, since in most practical situations the steady-state output error, due to a unit step input, will be required to be zero.

The development differs slightly depending on whether the dimension of the model, ℓ, is less than or equal to the system's dimension.

Consider first the case where $\ell = n$. The error equation, written out in detailed form, is in this case:

$$
\begin{bmatrix} \Delta\dot{x}_1 \\ \Delta\dot{x}_2 \\ \vdots \\ \Delta\dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0\ldots0 \\ 0 & 0 & 1 & 0\ldots0 \\ \vdots & & & \vdots \\ 0 & & \ldots & 1 \\ -\alpha_0 & -\alpha_1 & \ldots & -\alpha_{n-1} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix} + \begin{bmatrix} 0 & \ldots & 0 \\ 0 & \ldots & \\ & & \vdots \\ \vdots & & \\ 0 & \ldots & 0 \\ (\alpha_0-a_0) & (\alpha_1-a_1)\ldots & (\alpha_{n-1}-a_{n-1}) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{3.79}
$$

where $\underline{a}^T = (a_0, a_1, \ldots, a_{n-1})$ and $\underline{\alpha}^T = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$ are the coefficients of the system's and the model's characteristic equations, respectively.

There are two potential sources of excitation for this equation as seen in Figure 3.1. First, any initial error will result in an error response. For systems without zeros in the transfer function this effect does not have to be considered, since the initial error state is zero as seen above. The second source is the scalar input:

$$
i(t) = \sum_{i=0}^{n-1} (\alpha_i - a_i)\, x_{i+1} = (\underline{\alpha} - \underline{a})^T \underline{x} \tag{3.80}
$$

which is the only forcing term in Equation (3.79).

Since the model coefficient matrix $\hat{\underline{A}}$ is a specified constant matrix the error response can only be influenced by changing the input to the equations. It is clear, for instance, that the error response is equal to zero for all time when $i(t) = 0$ as there is no disturbance to the error state equation in this case. Thus one obvious way of reducing the error between the response of the system and the model is to minimize some measure of the input excitation to the error equations which, fortunately, happens to be a scalar when the equations are written in the phase-variable form. One such measure is the time integral over all time of the square of this input:

$$J = \int_0^{\infty} \underline{x}^T (\underline{\alpha}-\underline{a})(\underline{\alpha}-\underline{a})^T \underline{x} \, dt \qquad 3.81)$$

The weighting matrix in the quadratic cost functional has thus been determined as:

$$\underline{Q} = (\underline{\alpha}-\underline{a})(\underline{\alpha}-\underline{a})^T \qquad (\ell = n) \qquad (3.82)$$

This cost function can then be minimized with respect to the specified design parameters. Note, that $\underline{Q}$ is a function of $\underline{a}$ when $\ell=n$, which makes it a function of the system design parameters.

Consider now the case where the model's order is less than that of the system, i.e. $\ell<n$. The model equations are augmented and the error state equation becomes:

$$\begin{bmatrix} \Delta\dot{x}_1 \\ \vdots \\ \Delta\dot{x}_\ell \\ \vdots \\ \Delta\dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0\ldots0 & 0\ldots0 \\ \vdots & & \vdots & \vdots \quad \vdots \\ 0 & \ldots & 1 & \\ -\alpha_0 & -\alpha_1 & -\alpha_{\ell-1} & 0\ldots0 \\ 0 & \ldots & 0 & 0\ldots0 \\ \vdots & & \vdots & \vdots \quad \vdots \\ 0 & \ldots & 0 & 0\ldots0 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_\ell \\ \vdots \\ \Delta x_n \end{bmatrix} + \begin{bmatrix} 0 & 0 \ldots & 0 & 0 \ldots 0 \\ \vdots & & \vdots & \vdots \quad \vdots \\ 0 & \ldots & 0 & 0 \ldots 0 \\ \alpha_0 & \ldots & \alpha_{\ell-1} & 1\ 0\ldots0 \\ 0 & \ldots & & 0\ 1\ldots0 \\ \vdots & & & \vdots \\ 0 & \ldots & & 0\ 0 \quad 1 \\ -a_0 & -a_1 & \ldots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_\ell \\ \vdots \\ x_n \end{bmatrix}$$

(3.83)

It may be observed from the homogeneous part of this equation that the higher order error states, $\Delta x_{\ell+1} \ldots \Delta x_n$, do not affect the lower order error response directly. In fact the only excitation of the first ℓ error states is the scalar input:

$$i(t) = \sum_{i=0}^{\ell-1} (\alpha_i x_i) + x_{\ell+1} = \underline{\tilde{\alpha}}^T \underline{x} \tag{3.84}$$

where $\underline{\tilde{\alpha}}$ is an n-dimensional vector defined by:

$$\underline{\tilde{\alpha}}^T = [\alpha_0, \ldots, \alpha_{\ell-1}, 1, 0, \ldots, 0]$$

It is seen from Equation (3.83) that the responses of the first ℓ states are identical for the system and the model when $i(t) = 0$, since in this case the first ℓ error states are undisturbed. Actually this could only occur when the last $(n-\ell)$ system states are zero for all time and therefore identical to the corresponding model states.

Because the model, in reality, only specifies the desired response of the first ℓ state variables of the system, there is no need to constrain the response of the last $n-\ell$ states except for its influence on the lower order states. This influence is represented by the term $x_{\ell+1}$ in the input excitation to the lower order error response.

The cost function is now formed as before:

$$J = \int_0^\infty \underline{x}^T \underline{\tilde{\alpha}} \, \underline{\tilde{\alpha}}^T \underline{x} \, dt \tag{3.85}$$

and the state variable weighting matrix is:

$$\underline{Q} = \underline{\tilde{\alpha}} \, \underline{\tilde{\alpha}}^T \qquad (\ell < n) \tag{3.86}$$

It has been shown above that the model performance index can be interpreted as a quadratic measure of the scalar forcing term of the error state equation. It is interesting to note that all the terms making up this input term are dimensionally consistent. The units of $i(t)$ must be the same as those of the derivative of the $\ell^{\underline{th}}$ error state as can be seen from Equation (3.83). If this input is regarded as generalized "power", the model performance index is a measure of the "energy" driving the error equation.

The sensitivity term in the expected value of J can also be given a simple interpretation in terms of the input to the error model when $\ell<n$. The variation of i(t) can be obtained in this case from Equation (3.84) as:

$$\delta i(t) = \underline{\tilde{\alpha}}^T \, \delta \underline{x} \tag{3.87}$$

Using $\delta i^2(t)$ as an integrand in a sensitivity index gives:

$$\int_0^\infty \delta i^2(t) \, dt = \int_0^\infty \delta \underline{x}^T \, \underline{\tilde{\alpha}} \, \underline{\tilde{\alpha}}^T \, \delta \underline{x} \, dt = \int_0^\infty \delta \underline{x}^T \, \underline{Q} \, \delta \underline{x} \, dt \tag{3.88}$$

whose expected value is identical to the second term of Equation (3.23).

3.5.3 Systems with Zeros

The effect of zeros in the transfer function is represented by the last m initial states of the transient response as is seen from Equation (3.11). The corresponding initial error states are therefore non-zero, in general, and must be considered as a disturbance to the error equation in addition to the excitation input term. Rediess [31] solved this problem by adding a quadratic term in the initial error state to the performance index, such that:

$$J = \Delta \underline{x}_0^T \, \underline{W} \, \Delta \underline{x}_0 + \int_0^\infty \underline{x}^T \, \underline{Q} \, \underline{x} \, dt \tag{3.89}$$

In this way both sources of excitation to the error equation have been included in the performance index which is then minimized as before. The weighting matrix, $\underline{W}$, is a positive definite matrix which determines the relative importance of the initial error states as well as the weighting of this term relative to the integral term. Within the constraint of being positive definite,its choice is arbitrary. This presents a difficulty since there is no simple method for determining the relative effects of these terms.

A different approach will be taken here which eliminates the need for including the initial error states in the performance index at the cost of some restrictions in the choice of the model representing the desirable response. It is convenient to use the transfer functions of the system and the model for this derivation rather than their state-space realizations.

It was seen in the previous subsection that the error response of systems without transfer zeros can be obtained by passing an appropriate input signal through the model equations. Thus, it is assumed that the error response of any system can be obtained as shown in Figure 3.2. The output error Δy is equal to the corresponding transient error Δx, since the steady-state outputs of the system and the model are taken to be equal.

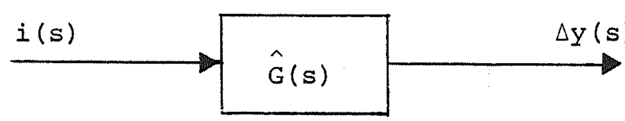


Figure 3.2 Error Response

The error response can be written as:

$$\Delta y(s) = [G(s) - \hat{G}(s)]\, u(s) \tag{3.90}$$

where $G(s)$ and $\hat{G}(s)$ are the transfer functions of the system and the model, respectively, and $u(s)$ is the transform of the step input in this case. From Figure (3.2) we have that:

$$\Delta y(s) = \hat{G}(s)\ i(s) \tag{3.91}$$

These two equations give the following expression for i(s):

$$i(s) = \left[\frac{G(s)}{\hat{G}(s)} - 1\right] u(s) \tag{3.92}$$

In the case of systems and models without zero i(s) can be written as:

$$i(s) = \frac{1}{\beta_0}(s^{\ell}+\alpha_{\ell-1}s^{\ell-1}+\ldots+\alpha_1 s+\alpha_0)\ y(s) - u(s) \tag{3.93}$$

where the transfer function of the model, $\hat{G}(s)$, is given by:

$$\hat{G}(s) = \frac{\beta_0}{s^{\ell} + \alpha_{\ell-1}s^{\ell-1} + \ldots + \alpha_1 s + \alpha_0} \tag{3.94}$$

Transforming i(s) into the time domain gives:

$$i(t) = \frac{1}{\beta_0}\left[\alpha_0 y(t) + \alpha_1 \dot{y}(t) + \ldots + \alpha_{\ell-1} y^{(\ell-1)}(t) + y^{(\ell)}(t)\right] - 1 \tag{3.95}$$

since u(t) is the unit step function and the system is at rest initially. From Equation (3.1) we have that for a stable system:

$$\lim_{t\to\infty} y(t) = \frac{b_0}{a_o}$$

which in turn can be used to show that:

$$\lim_{t\to\infty} i(t) = \frac{\alpha_0}{\beta_0}\frac{b_0}{a_0} - 1 = 0$$

as all the derivatives of y(t) must go to zero in steady-state and

$\frac{\alpha_0}{\beta_0}\frac{b_0}{a_0} = 1$ by the assumption that the static sensitivities of the system and model are equal. Equation (3.95) can then be written in terms of the transient response as:

$$i(t) = \frac{1}{\beta_0}\,[\alpha_0 x(t) + \alpha_1 \dot{x}(t) + \ldots + \alpha_{\ell-1} x^{(\ell)}(t) + x^{(\ell+1)}(t)] \qquad (3.96)$$

For $\ell<n$, i.e. a model of lower order than the system, i(t) can be written in the form:

$$i(t) = \frac{1}{\beta_0}\,\underline{\tilde{\alpha}}^T\,\underline{x}(t) \qquad (3.97)$$

where $\underline{x}(t)$ is the system state vector as described by Equation (3.5). This expression for i(t) is identical to the forcing term in the error equation (3.83) except for the division by the constant, β_0. The performance index for $\ell<n$ is then defined by:

$$J = \frac{1}{\beta_0^2}\int_0^\infty \underline{x}^T(t)\,\underline{\tilde{\alpha}}\,\underline{\tilde{\alpha}}^T\,\underline{x}(t)\,dt \qquad (3.98)$$

which is identical to the performance index as defined by Equation (3.85), except for the constant factor.

Consider now the case when the model is of the same order as the system, i.e. $\ell=n$. Equation (3.96) can be written in terms of the system state vector by substituting the following expression for $x^{(n)}(t)$:

$$x^{(n)}(t) = \dot{x}_n = -\underline{a}^T\,\underline{x} \qquad (3.99)$$

which is obtained from the state equation (3.5). i(t) then becomes:

$$i(t) = \frac{1}{\beta_0}\,(\underline{\alpha} - \underline{a})^T\,\underline{x} \qquad (3.100)$$

which is proportional to the input to the error equation (3.79) for this case. The resulting performance index is also equivalent to that of Equation (3.81). This approach, therefore, leads to the same result as was obtained in Section 3.5.2 for systems without zeros.

Consider now the case of systems and equations with zeros in their transfer functions. Equation (3.92) can now be written as:

$$i(s) = \left[\frac{s^{\ell}+\alpha_{\ell-1}s^{\ell-1}+\ldots+\alpha_1 s+\alpha_0}{\beta_k s^k+\ldots+\beta_1 s+\beta_0} \cdot \frac{b_m s^m+\ldots+b_1 s+b_o}{s^n+a_{n-1}s^{n-1}+\ldots+a_1 s+a_0} -1 \right] u(s) \tag{3.101}$$

This equation can be written in the form of Equation (3.93) by defining a new system, whose transfer function, $\tilde{G}(s)$, contains the zeros of the model as system poles in addition to the regular system transfer function:

$$\tilde{G}(s) = \frac{\beta_0(b_m s^m+\ldots+b_1 s+b_0)}{(\beta_k s^k+\ldots+\beta_1 s+\beta_0)(s^n+a_{n-1}s^{n-1}+\ldots+a_1 s+a_0)} \tag{3.102}$$

This expanded system is of $(n+k)^{\underline{th}}$ order and it should be noted that the new system poles are cascaded to the original closed-loop transfer function. Hence, they do not affect the behavior of the closed-loop transfer function directly.

Thus, the zeros have been removed from the model and cascaded as poles to the system transfer function. The output of this new system is expressed by:

$$\tilde{y}(s) = \tilde{G}(s)\ u(s)$$

Using this result, Equation (3.101) can be written as:

$$i(s) = \left[\frac{1}{\beta_0}\left(s^{\ell} + \alpha_{\ell-1}s^{\ell-1} + \ldots + \alpha_1 s + \alpha_0\right)\tilde{y}(s) - u(s)\right] \tag{3.103}$$

For $\ell \leq \big((n+k)-m\big)$ the time domain version of this equation is:

$$i(t) = \frac{1}{\beta_0}\left(\alpha_0\tilde{y}(t) + \alpha_1\dot{\tilde{y}}(t) + \ldots + \alpha_{\ell-1}\tilde{y}^{(\ell-1)}(t) + \tilde{y}^{(\ell)}(t)\right) - 1 \tag{3.104}$$

This follows from the fact that the first n-m-1 derivatives of the output, in a system with n poles and m zeros, are zero at $t=0^+$ for a step input. This can be verified from the system equations of Section 2.2. As a result:

$$\mathcal{L}\left[y^{(i)}(t)\right] = s^i\, y(s) \qquad 0 \leq i \leq n-m \tag{3.105}$$

In the expanded system, n-m is simply replaced by n+k-m since k poles have been added to the system equation. Equation (3.104) is then written in terms of the transient response:

$$i(t) = \frac{1}{\beta_0}\left(\alpha_0\tilde{x}(t) + \alpha_1\dot{\tilde{x}}(t) + \ldots + \alpha_{\ell-1}\tilde{x}^{(\ell-1)}(t) + \tilde{x}^{(\ell)}(t)\right) \tag{3.106}$$

where the static sensitivities of the system and the model are taken to be equal as before.

For a model, whose number of excess poles over zeros is equal to or less than that of the system, i.e. $\ell-k \leq n-m$, i(t) can be written:

$$i(t) = \frac{1}{\beta_0}\,\underline{\tilde{\alpha}}^T\,\underline{\tilde{x}}(t) \tag{3.107}$$

where $\underline{\tilde{x}}(t)$ is the state vector of the expanded system equations in the homogeneous phase-variable form, which consists of the transient output response and its derivatives. For ℓ-k$\leq$n-m, the performance index is:

$$J = \frac{1}{\beta_0^2} \int_0^\infty \underline{\tilde{x}}^T \, \underline{\tilde{\alpha}} \, \underline{\tilde{\alpha}}^T \, \underline{\tilde{x}} \, dt \tag{3.108}$$

with the state vector, $\underline{\tilde{x}}(t)$, described by:

$$\dot{\underline{\tilde{x}}} = \underline{\tilde{A}} \, \underline{\tilde{x}} \; ; \; \underline{\tilde{x}}(0) = \underline{\tilde{x}}_0 \tag{3.109}$$

where $\underline{\tilde{A}}$ is the coefficient matrix of the expanded system and the initial condition is given by Equation (3.11), substituting the coefficients of the extended system equations.

When the model has more excess poles over zeros than the system, i.e. ℓ-k>n-m, some care must be exercised in transforming Equation (3.103) into the time domain. Consider, for instance, the case when ℓ-k = n-m+1. The transform of $\tilde{y}^{(\ell)}(t)$ is now:

$$\mathcal{L}[\tilde{y}^{(\ell)}(t)] = s^\ell \, \tilde{y}(s) - \tilde{y}^{(\ell-1)}(0+) \tag{3.110}$$

since ℓ-1 = n+k-m and $\tilde{y}^{(n+k-m)}(0+)$ is non-zero, in general, as can be seen from the (n+k-m)$^{\underline{th}}$ row of the state equation (3.2) for the expanded system:

$$\tilde{y}^{(n+k-m)}(0+) = \dot{y}_{n+k-m}(0+) = c_{n+k-m} \tag{3.111}$$

where the system is at rest initially. The following result can then be obtained from Equation (3.110) by inverse transformation:

$$\mathcal{L}^{-1} [s^\ell \, \tilde{y}(s)] = \tilde{y}^{(\ell)}(t) + \delta(t) \, \tilde{y}^{(\ell-1)}(0+) \tag{3.112}$$

where $\delta(t)$ is the Dirac delta function. $i(t)$ would, therefore, contain the delta function in this case, which is unacceptable. Higher values of $(\ell-k)$ result in even more complicated expressions for $i(t)$. For this reason, the model will be required to satisfy the condition that $(\ell-k) \leq (n-m)$. This is not a serious restriction, since it is not clear that anything is gained by using a model with more excess poles over zeros than the system.

To summarize, it has been shown that the difficulties associated with defining the model performance index, when the system and model contain zeros, can be avoided by restricting the model from having more excess poles over zeros than the system. Furthermore, any model zeros are removed from the model transfer function and added as cascaded poles to the system transfer function. This transformation of the problem has been shown to be consistent with the definition of the model performance index in terms of the excitation input to the error model.

The technique of adding the model zeros as poles to the system can be given a simple interpretation, when the model and the system contain an equal number of poles and zeros. If it is assumed, furthermore, that complete matching of the model and system responses can be achieved, the model zeros represent desired locations of the system zeros. When the model zeros are added as fixed cascaded poles to the system, the resulting model contains only poles. When the performance index is minimized to obtain complete matching of the system and model responses, the system zeros must be moved such as to cancel with the new system poles at the same time as the original system poles become identical to the model poles. But this is the same as matching the system zeros and poles to those of the model, which is the desired result.

For systems without zeros, the weighting matrix of the model performance index becomes a function of the characteristic coefficients when a model of equal order is used, as can be seen from Equation (3.81). This is a computational inconvenience, particularly when considering the expected value of J due to parameter variations, which can be avoided by using the technique of expanding the system equations. Thus by adding a cancelling pole and zero pair at some convenient location to the system transfer function, the system order has been increased by one over the model, but the number of excess poles over zeros is equal for both. The method described in this subsection can then be applied, resulting in a constant weighting matrix for the performance index.

It should be noted that the expansion of the system is achieved by adding singularities to the transfer function after all loops have been closed. These new singularities, therefore, do not affect the loci of the closed-loop roots.

More detailed information about the model performance index can be obtained from References [31] and [32].

3.6 The ISE Performance Index

The integral square error performance index can be defined as a quadratic expression in terms of the state vector of the error model in Figure 3.2:

$$J = \int_0^\infty \Delta\underline{x}^T \, \underline{Q} \, \Delta\underline{x} \, dt \qquad (3.113)$$

where $\Delta\underline{x}$ is defined as the difference between the system and model responses. The difference between the model performance index and the ISE index is that the MPI focuses on the input to the error model whereas the ISE index is defined in terms of the output of the error model and is consequently a more direct measure of the difference between the system and model responses than the MPI.

The disadvantage of the integral square error index is that it contains the model response explicitly, which increases the computational task. In addition, there is no systematic way of selecting the weighting matrix in Equation (3.113). In most cases only the first element of $\Delta\underline{x}$ has been used, i.e. the scalar output error. Addition of derivatives of the output error can then be made on a trial and error basis.

The equations derived in Sections 3.3 and 3.4 must be augmented in order to accommodate the integral square error index. First, Equation (3.113) can be written:

$$J = \int_0^\infty [\underline{x}^T \underline{Q}\, \underline{x} - 2\, \underline{x}^T \underline{Q}\, \hat{\underline{x}} + \hat{\underline{x}}^T \underline{Q}\, \hat{\underline{x}}]\, dt \tag{3.114}$$

since $\underline{Q}$ is a symmetric matrix. If the model is of lower order than the system, its state vector can be augmented to $n^{\underline{th}}$ dimension by the addition of zeros. Taking the expected value of J gives the following result:

$$\bar{J} = \int_0^\infty [\underline{x}_*^T Q\, \underline{x}_* - 2\, \underline{x}_*^T \underline{Q}\, \hat{\underline{x}} + \hat{\underline{x}}^T \underline{Q}\, \hat{\underline{x}} + \overline{\delta\underline{x}^T \underline{Q}\, \delta\underline{x}}\,]\, dt \tag{3.115}$$

since the nominal response of the system is also its expected value and the model response is deterministic. By comparison with Equation (3.23) it is seen that the expected value of the ISE index contains two additional terms, both of which are deterministic. These terms can be written as:

$$\int_0^\infty (\hat{\underline{x}}^T \underline{Q}\, \hat{\underline{x}} - 2\, \underline{x}_*^T \underline{Q}\, \hat{\underline{x}})\, dt = \mathrm{tr}\left[\underline{Q}\, [\hat{\underline{X}} - 2\hat{\underline{Y}}]\right] \tag{3.116}$$

where

$$\underline{\hat{X}} = \int_0^{\infty} \underline{\hat{x}}\ \underline{\hat{x}}^T\ dt \qquad \text{and} \qquad \underline{\hat{Y}} = \int_0^{\infty} \underline{\hat{x}}\ \underline{x}_*^T\ dt$$

Assuming that the equations of the system and the model are in the standard observable form of Section 3.2, the following equations are obtained for $\underline{\hat{X}}$ and $\underline{\hat{Y}}$ using the approach of Section 3.3:

$$\underline{\hat{A}}\ \underline{\hat{X}} + \underline{\hat{X}}\ \underline{\hat{A}}^T + \underline{\hat{X}}_0 = \underline{0}$$

$$\underline{A}\ \underline{\hat{Y}} + \underline{\hat{Y}}\ \underline{\hat{A}}^T + \underline{\hat{Y}}_0 = \underline{0}$$

where

$$\underline{\hat{X}}_0 = \underline{\hat{x}}_0 \underline{\hat{x}}_0^T \qquad \text{and} \qquad \underline{\hat{Y}}_0 = \underline{x}_0 \underline{\hat{x}}_0^T$$

$\underline{A}$ and $\underline{\hat{A}}$ are the system and model coefficient matrices, respectively, as before and $\underline{\hat{Y}}$ is an $n \times \ell$ matrix. The corresponding terms in the necessary conditions for a minimum of $\overline{J}$ must also be determined. The equation for $\underline{\hat{X}}$ is only dependent on the model response and is, consequently, not a function of the free design parameters. There is, therefore, no need to adjoin Equation (3.117) with Lagrange multipliers to $\overline{J}$, since it is always satisfied despite variations of the free design parameters.

Equation (3.118), on the other hand, is a function of these parameters and must be adjoined to $\overline{J}$. The following term is then added to the performance index:

$$\mathrm{tr}\left\{\underline{P}_3\ [\underline{A}\ \underline{\hat{Y}} + \underline{\hat{Y}}\ \underline{\hat{A}}^T + \underline{\hat{Y}}_0]\right\} \tag{3.119}$$

where $\underline{P}_3$ is a matrix of Lagrange multipliers which by taking the variation of $\bar{J}$ with respect to $\hat{\underline{Y}}$ is found to satisfy the equation:

$$\underline{P}_3\underline{A} + \hat{\underline{A}}^T\underline{P}_3 = 2\underline{Q} \tag{3.120}$$

The variation of the term (3.119) with respect to $\underline{A}$ and $\hat{\underline{Y}}_0$ result in the following terms, which must be added to Equation (3.74), expressing the variation of $\bar{J}$ when all the constraining equations are satisfied:

$$\mathrm{tr}\left\{\underline{P}_3^T\hat{\underline{Y}}^T\tilde{\delta}\underline{A}^T+\underline{P}_3\tilde{\delta}\hat{\underline{Y}}_0\right\} = \left[-\underline{\eta}^T P_3^T\hat{\underline{Y}}^T\left[\frac{\partial \underline{a}}{\partial \underline{p}}\right] + \hat{\underline{x}}_0^T\ \underline{P}_3\left[\frac{\partial \underline{x}_0}{\partial \underline{p}}\right]\right]\delta\underline{p} \tag{3.121}$$

where $\underline{\eta}$ was defined as an n-dimensional vector:

$$\underline{\eta}^T = [\ 0,0,\ldots,0,1\ ]$$

Thus, all the equations for computing the expected value of the integral square error index have been determined as well as the necessary conditions for a local minimum of its value. Specifically, it is now necessary to find the solution of two additional matrix equations in order to determine the value of the ISE index as compared with the performance index containing only the system response. The problem of computing the minimum value of $\bar{J}$ and the corresponding free design parameter values is discussed in detail in Chapter 4.

3.7 Example Problem

A simple example, which shows the effect of using the expected value of the integral square error performance index, will be given.

Example 3.1

The plant consists of a simple integrator with variable static sensitivity whose nominal value can be chosen. The desired response is represented by the step response of a first order model with the transfer function:

$$\hat{G}(s) = \frac{1}{s + 1} \tag{3.122}$$

An identical nominal response can be obtained for the system by adding a feedback path around the plant as shown in Figure 3.3 and choosing the nominal open-loop static sensitivity equal to unity as:

$$G(s) = \frac{S_{OL}}{s + S_{OL}} \tag{3.123}$$

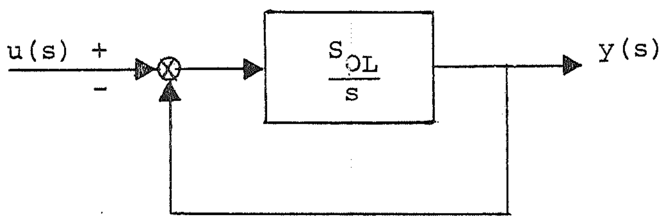


Figure 3.3 System with Unity Feedback

This system is relatively sensitive to changes in S_{OL} as can be seen, for instance, from the variation of its pole position, which is given by:

$$\delta p_1 = -\delta S_{OL} = -S^*_{OL}\left(\frac{\delta S_{OL}}{S^*_{OL}}\right) \tag{3.124}$$

Any change in S_{OL}, therefore, results in an equal change in p_1. In order to decrease the sensitivity a zero may be added to the feedback path as shown in Figure 3.4.

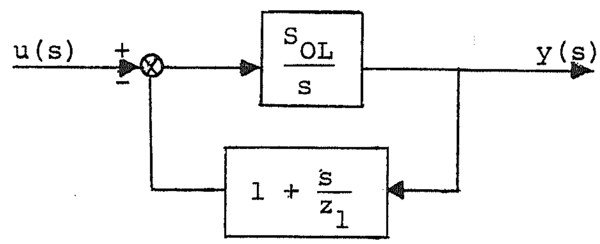


Figure 3.4 System with zero in Feedback

The transfer function of the system is given by:

$$G(s) = \frac{\dfrac{S_{OL} z_1}{S_{OL} + z_1}}{s + \dfrac{S_{OL} z_1}{S_{OL} + z_1}} \tag{3.125}$$

where S_{OL} and z_1 are free to be chosen. The nominal response of the system can be made identical to the model response by choosing:

$$a_0 = \frac{S_{OL} z_1}{S_{OL} + z_1} = 1 \tag{3.126}$$

in which case the variation of the system pole becomes:

$$\delta p_1 = - \frac{1}{S^*_{OL}} \left(\frac{\delta S_{OL}}{S^*_{OL}} \right), \tag{3.127}$$

Equation (3.126) determines an infinite number of combinations of the design parameters which result in a nominal response identical to that of the model. The addition of the zero in the feedback path, therefore, provides increased freedom in the design, which is not required in order to satisfy the requirements on the nominal response, but could possibly be used to reduce the effect of changes in the static sensitivity on the system response. From Equation (3.127) it is observed that, for a specified percentage change in S_{OL}, the resulting variation of the pole can be made arbitrarily small by choosing a large nominal value of S_{OL}.

The ISE index for this first order problem can be written:

$$J = \int_0^\infty (x-\hat{x})^2\, dt = \int_0^\infty x^2\, dt + \int_0^\infty \hat{x}^2\, dt - 2\int_0^\infty x\,\hat{x}\, dt \tag{3.128}$$

where x and $\hat{x}$ are the transient responses of the system and model to a unit step input, respectively. These integrals can be determined in terms of the system and model coefficients by solving Equations (3.27), (3.117) and (3.118), which are scalar equations for a first order system. Thus, we have that:

$$\begin{aligned} a_0x_{11} + a_0x_{11} &= x_0^2 = 1 \quad ; \quad x_0 = -1 \\ \alpha_0\hat{x}_{11} + \alpha_0\hat{x}_{11} &= \hat{x}_0^2 = 1 \quad ; \quad \hat{x}_0 = -1 \\ a_0\hat{y}_{11} + \alpha_0\hat{y}_{11} &= x_0\hat{x}_0 = 1 \end{aligned} \tag{3.129}$$

where x_{11}, $\hat{x}_{11}$ and $\hat{y}_{11}$ denote the three integrals in Equation (3.128), which by substitution of their solutions becomes:

$$J = \frac{(a_0-\alpha_0)^2}{2a_0\alpha_0(a_0+\alpha_0)} \tag{3.130}$$

Clearly, J=0 only when the transfer function of the system is identical to that of the model, i.e. $a_0 = \alpha_0$. The expected value of J can now be written as:

$$\overline{J} = \frac{(a_0^* - \alpha_0)^2}{2a_0^* \, \alpha_0 (a_0^* + \alpha_0)} + \int_0^\infty \overline{\delta x^2} \, dt \tag{3.131}$$

where the nominal values are denoted by an asterisk and δx is the variation of the system response to changes in S_{OL}. The value of the integral term can be determined by solving Equations (3.34), (3.40) and (3.43) for this problem, recognizing that $\delta x_0 = 0$, since the closed-loop static sensitivity is always equal to unity:

$$\begin{aligned} a_0 \, \overline{\delta x_{11}} + a_0 \, \overline{\delta x_{11}} &= -2 \, z_0 \\ a_0 \, z_0 \quad + a_0 \, z_0 \quad &= -x_{11} \omega \end{aligned} \tag{3.132}$$

where $\overline{\delta x_{11}}$ denotes the integral term and $\omega = \overline{\delta a_0^2}$.

The solution of these equations, using the previously obtained solution for x_{11}, gives the following result:

$$\overline{\delta x_{11}} = \frac{\overline{\delta a_0^2}}{4a_0^3} \tag{3.133}$$

Assuming that the response of the system is constrained to be identical to the model's response for the nominal value of S_{OL}, the first term of $\overline{J}$ is zero. $\overline{J}$ is then equal to the second term, which represents the effect of the parameter variation on the expected value of the performance index. For the first configuration with unity feedback, this term becomes:

$$\overline{\delta x_{11}} = \frac{1}{4} S_{OL}^{*2} \overline{\left(\frac{\delta S_{OL}}{S_{OL}^*}\right)^2} = \frac{1}{4} \left(\frac{\delta S_{OL}}{S_{OL}^*}\right)^2 \tag{3.134}$$

since $a_0 = S_{OL}$ and $a_0 = 1$. The expression for the second configuration is similarly obtained as:

$$\overline{\delta x_{11}} = \frac{1}{4S_{OL}^{*2}} \overline{\left(\frac{\delta S_{OL}}{S_{OL}^{*}}\right)^2} \tag{3.135}$$

It is reasonable to take the mean square value of the percentage variation in S_{OL} to be constant, since this variation is unlikely to be affected by the choice of the nominal value. This nominal value is fixed for the unity feedback system and there is no way of reducing the value of $\overline{\delta x_{11}}$ without changing the output response. The addition of a zero to the feedback allows S_{OL} to be chosen freely, with the value of z_1 determined by Equation (3.126) under the assumption of perfect model following. With $\overline{\delta x_{11}}$ inversely proportional to the square of the static sensitivity, it is clear that S_{OL}^* would be chosen as large as possible. As S_{OL} approaches infinity we have from Equation (3.126) that:

$$\lim_{S_{OL}\to\infty} z_1 = \lim \frac{S_{OL}}{S_{OL}-1} = 1 \tag{3.136}$$

Hence, the expected value of the integral square error index is minimized, in theory, by placing the feedback zero at the desired pole location and using infinitely large gain. In practice the available gain is limited and S_{OL}^* would be chosen to be at its upper limit.

The response of the system with the unity feedback is shown in Figure 3.5 for the nominal value of S_{OL} as well as for $\pm 50\%$ variation from that value. The corresponding responses of the system with a transfer zero in the feedback path are shown in Figure 3.6, where S_{OL}^* =3, which is arbitrarily chosen as the upper limit of that value. The corresponding zero location is at $z_1 = -1.5$.

The nominal responses are identical by constraint, but it is clear that the deviations of the second system configuration to changes in the gain are considerably smaller than those of the

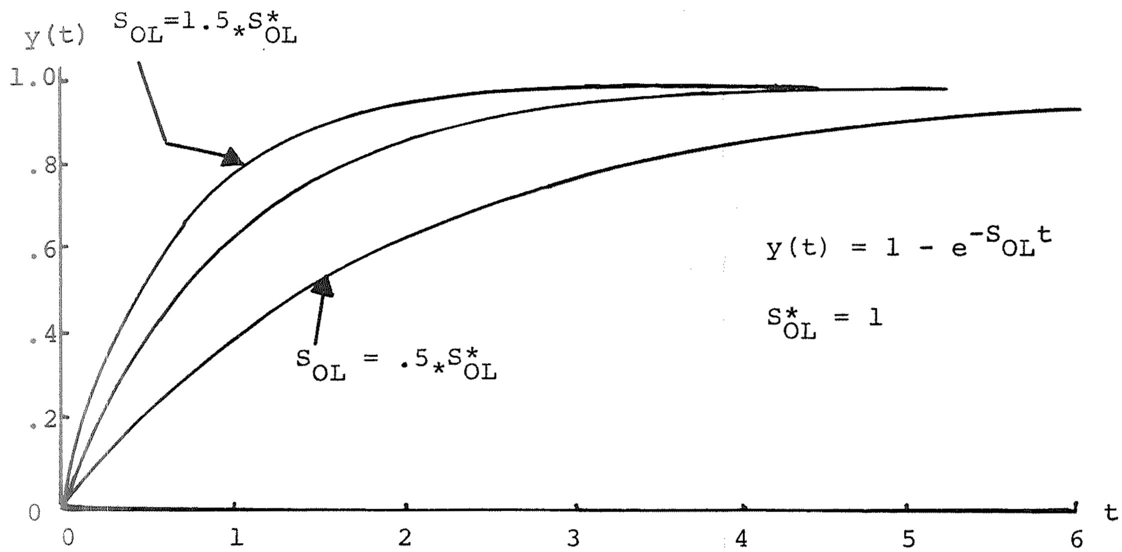


Figure 3.5. Response of system with unity feedback

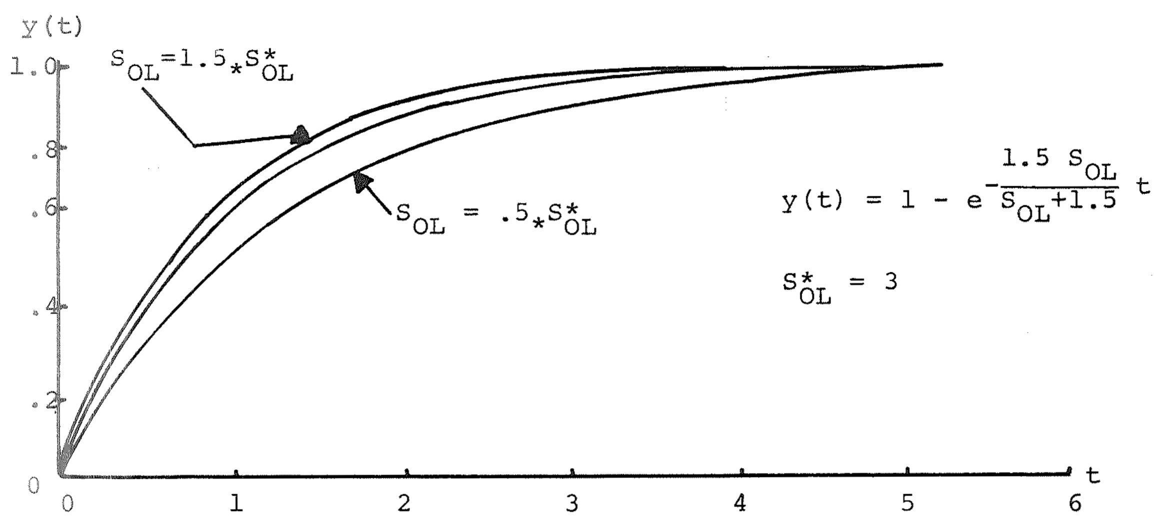


Figure 3.6. Response of system with zero in feedback

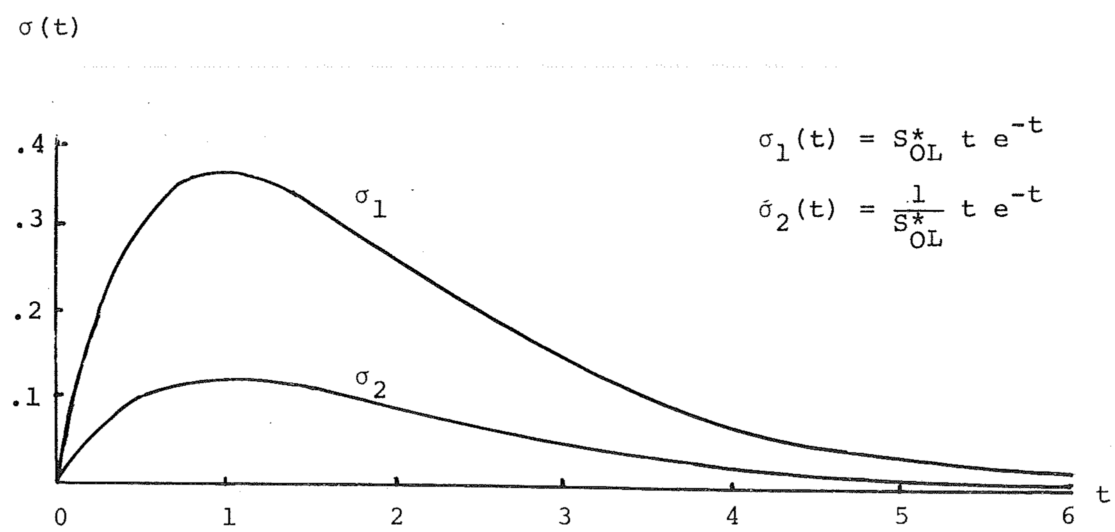


Figure 3.7. Sensitivity functions of system configurations

system with unity feedback. Thus, the response envelope in Figure 3.6, which is determined by the high and low values of S_{OL}, falls completely within the corresponding envelope in Figure 3.5. The sensitivity functions of the two configurations are plotted in Figure 3.7. These are proportional to the first order deviations of the system responses and are in fairly good agreement with the actual deviations.

The sensitivity of the first order system to changes in gain has been significantly reduced by the addition of a transfer zero to the feedback path to provide the additional freedom in the design and by using the expected value of the performance index as a guide in the selection of the design parameters. The nominal responses of the system were required to be identical to the model response in order to observe the effect of the second term in $\overline{J}$ separate from the nominal term. This would not be done in most applications unless there was a specific requirement on the nominal response.

The implicit assumption has been made here that the location of the feedback zero is absolutely stable. This is not necessarily the case and the effect of variations of z_1 on the solution will now be explored. The nominal responses are still required to be identical to that of the model. The design with unity feedback is independent of the zero variations and is, therefore, unaffected. The effect of the parameter variations on the second design is again expressed by the second term of the performance index, which in this case becomes:

$$\overline{\delta x_{11}} = \frac{1}{4S_{OL}^{*2}} \overline{\left(\frac{\delta S_{OL}}{S_{OL}^*}\right)^2} + \frac{1}{4z_1^{*2}} \overline{\left(\frac{\delta z_1}{z_1^*}\right)^2} \tag{3.137}$$

where it has been assumed that the two variations are uncorrelated. The restriction on the nominal performance gives the following relationship between S_{OL}^* and z_1^*:

$$z_1^* = \frac{S_{OL}^*}{S_{OL}^*-1} \tag{3.138}$$

which by substitution into Equation (3.137) gives:

$$\overline{\delta x_{11}} = \frac{1}{4S_{OL}^{*\,2}}\left[\overline{\left(\frac{\delta S_{OL}}{S_{OL}^*}\right)^2} + (S_{OL}^*-1)^2\,\overline{\left(\frac{\delta z_1}{z_1^*}\right)^2}\right] \tag{3.139}$$

This expression can now be minimized with respect to S_{OL}^* by setting its derivative equal to zero. The corresponding values of S_{OL}^* and z_1^* which are the design values, are:

$$S_{OL}^* = 1 + \frac{\overline{\left(\frac{\delta S_{OL}}{S_{OL}^*}\right)^2}}{\overline{\left(\frac{\delta z_1}{z_1^*}\right)^2}}; \quad z_1^* = 1 + \frac{\overline{\left(\frac{\delta z_1}{z_1^*}\right)^2}}{\overline{\left(\frac{\delta S_{OL}}{S_{OL}^*}\right)^2}} \tag{3.140}$$

It may be noted that as the mean square value of the relative zero variation becomes small, S_{OL}^* approaches infinity, which is the solution obtained previously for a perfectly stable zero. When this mean square value becomes infinitely large, on the other hand, S_{OL}^* approaches unity and the nominal location of the zero becomes infinite. But this is identical to the solution with unity feedback. This result indicates that the zero in the feedback can always be used to reduce the sensitivity of the system, as defined by the term $\overline{\delta x_{11}}$, regardless of the variations in its location. This can be shown by substituting the expression for the design value of S_{OL}^* into Equation (3.139):

$$\overline{\delta x_{11}} = \frac{1}{4\left[1 + \dfrac{\overline{\left(\dfrac{\delta S_{OL}}{S^*_{OL}}\right)^2}}{\overline{\left(\dfrac{\delta z_1}{z^*_1}\right)^2}}\right]} \overline{\left(\frac{\delta S_{OL}}{S^*_{OL}}\right)^2} \tag{3.141}$$

For the same gain variation this value is always smaller than the corresponding value for the unity feedback system, as given by Equation (3.134), although the reduction becomes insignificant when the zero variations become large. It is interesting to note that the amount of reduction in sensitivity achieved by adding the zero, depends only on the ratio of the mean square values of the relative parameter variations.

Finally, the integral square error index, as given by Equation (3.130), is plotted as a function of $\frac{S_{OL}}{S^*_{OL}}$ and $\frac{z_1}{z^*_1}$ in Figures 3.8 and 3.9, respectively, for a number of values of the parameter variation ratio. The nominal design values are given by Equation (3.140) as functions of this ratio. One of the parameters is held constant at its nominal value while the other is varied in order to determine the values of J.

In Figure 3.8, the system with unity feedback is represented by $\gamma=0$, since in this case the zero has infinite variation about its nominal and is removed from the system altogether. When this variation decreases relative to that of the static sensitivity and the effect of the feedback zero is increased, the curvature of the ISE index as a function of $\frac{S_{OL}}{S^*_{OL}}$ is decreased at the minimum. The effect of the percentage variation in S_{OL}, which is a specified constant, is therefore decreased and becomes zero in the limit when the feedback zero location is perfectly stable, i.e. $\gamma=\infty$. The opposite effect may be observed for the ISE index as a function of $\frac{z_1}{z^*_1}$ from Figure 3.9. As the effect

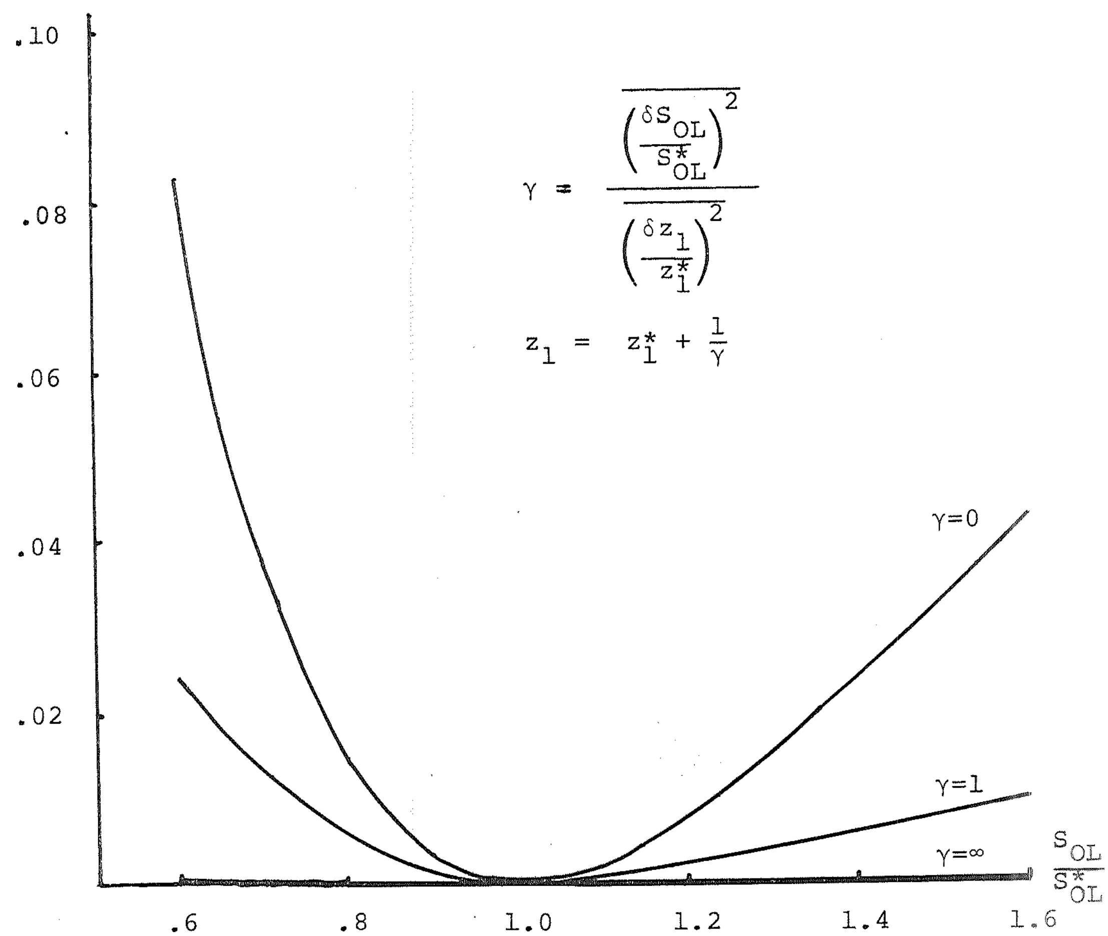


Figure 3.8. The ISE Index as a Function of S_{OL}

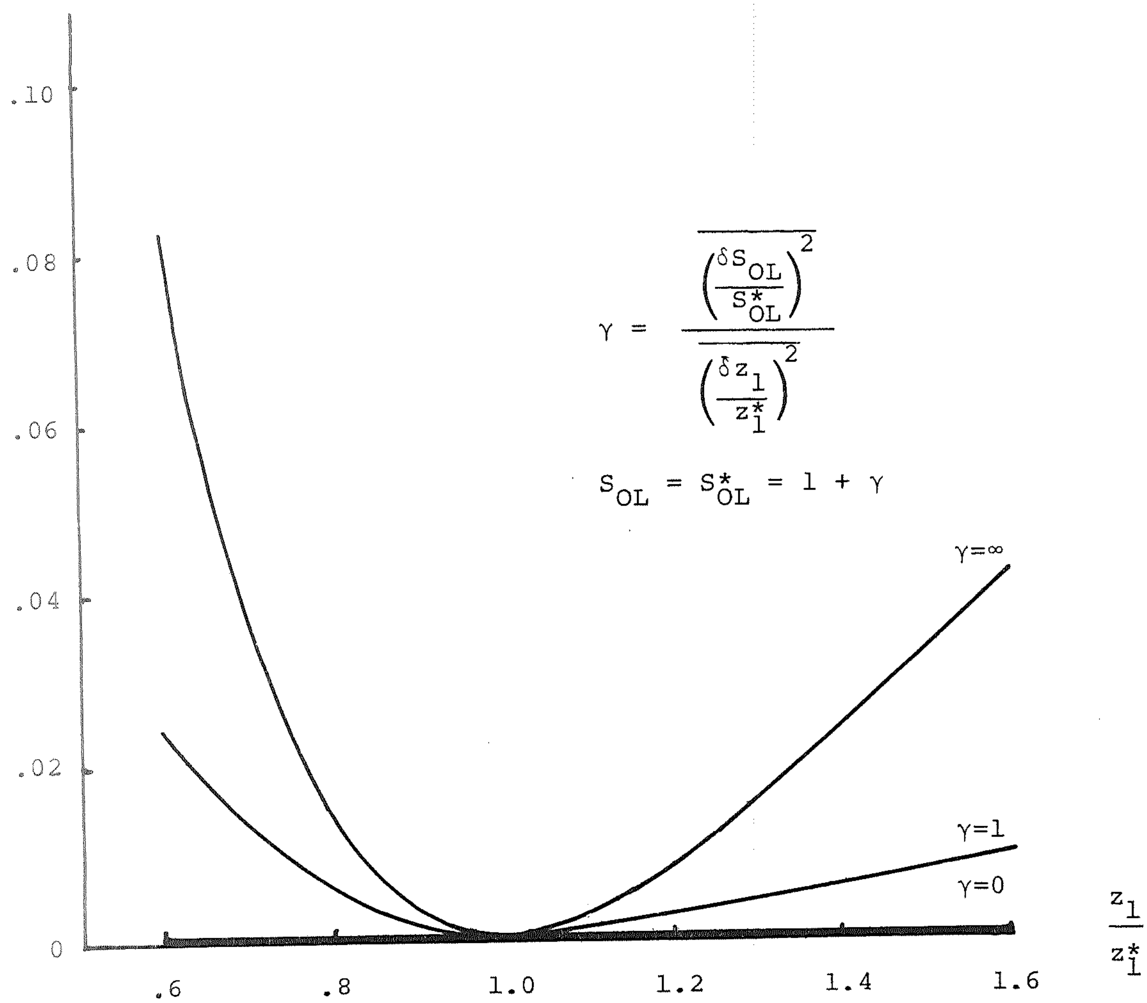


Figure 3.9. The ISE Index as a function of z_1

of the feedback zero is increased in order to decrease the system sensitivity to changes in S_{OL}, the curvature of the ISE index is increased. This clearly makes it more sensitive to a given percentage change in z_1 as the value of γ decreases, i.e. as the zero location becomes more stable. On the other hand, the actual variations of $\frac{z_1}{z^*_1}$ become smaller as γ decreases, since the percentage variation of S_{OL} is constant. The price paid for reducing the sensitivity of the system to changes in S_{OL}, hence, is not as high as might be expected from Figure 3.9.

The improvement in system sensitivity, as measured by the sensitivity index $\overline{\delta x_{11}}$ of Equation (3.137), is basically obtained by distributing the effect of parameter variations on the system between the two independent parameters according to their relative stability. Since it is statistically less likely that worst case conditions occur when two independent parameters are involved than in the case of a single parameter, an improvement is achieved as measured by the influence of the variations on the expected value of the performance index.

3.8 Multivariable Systems

So far only single input/output systems have been considered. As is well known, the various transfer functions relating the outputs to the inputs of a multivariable system all have the same poles, with the differences confined to the zeros. In many applications the system requirements make it necessary to specify the desirable characteristics of more than one of these input/output transfer functions. A straightforward approach to such problems is to treat them as separate but simultaneous problems. Thus, when the model responses of two such transfer functions are specified, a performance index can be formed for each of them. The simultaneous design process is then implemented, for instance, by minimizing the weighted sum of the two indices, which constitutes an overall system performance index. The emphasis on a given transfer function can then be varied by changing its corresponding weighting factor. The computational effort in finding the minimum of this new index is multiplied by the number of transfer functions considered, when compared with the effort required for a single input/output system. When only a few such transfer relationships have to be considered, such as is the case in most flight control systems, this is by no means an impractical task using the numerical methods described in Chapter 4.

A variation of this method consists of selecting mutually exclusive subsets of the design parameters with each set corresponding to a specified transfer function. The individual performance indices are then minimized one at a time varying only the appropriate subset of the design parameters in each case. The problem is, in effect, separated into a series of single input/output problems which must be solved in an iterative manner until a satisfactory result is achieved. It is not clear, however, that the computational task is any less in this case than is required for the minimization of the total performance index.

These methods are discussed in Reference [31] in terms of the model performance index, but other indices could be used as well.

3.9 Effects of Noise

The reduction of system sensitivity to parameter variations is very often obtained by significantly increasing the bandwidth of the feedback path beyond the bandwidth required by the nominal condition. This has the undesirable effect of amplifying sensor noise, for instance, which enters the system at the feedback level. It is, therefore necessary to give some consideration to how this effect can be taken into account in the design process.

A simple but practical approach is to estimate the maximum tolerable bandwidth from the knowledge of the power spectrum of the noise. This estimate can then be used to determine the permissible range of one or more of the design parameters. Numerous methods are available for constraining the values of these parameters.

A more systematic method, which is also compatible with the general design process, consists of defining a cost function representing the effects of the noise on the system output and adding it to the system performance index. One common function of this type is the mean value of a quadratic form of the system state vector, which is excited only by the noise input:

$$J = \overline{\underline{x}^T(t)\ \underline{Q}\ \underline{x}(t)} = \mathrm{tr}[\underline{Q}\ \overline{\underline{x}\ \underline{x}^T}] = \mathrm{tr}\ \underline{Q}\overline{\underline{X}} \tag{3.142}$$

where the weighting matrix may or may not be the same as the matrix used in the other terms of the performance index. The input noise is assumed to be Gaussian and can, therefore, be produced by passing uncorrelated white noise through an appropriate shaping filter. In the following derivation it is assumed that the transfer function of the shaping filter has already been determined and is included in the overall system transfer function. The state equation of the system is taken to be in the standard observable form of Section 3.2:

$$\dot{\underline{x}} = \underline{A}\ \underline{x} + \underline{c}\ u \tag{3.2}$$

where $\underline{A}$ is in the phase variable form as usual and $u(t)$ is the white noise input which is defined by:

$$\overline{u(t)\ u(T)} = \omega\ \delta(t-T) \tag{3.143}$$

where $\delta(t)$ is the unit delta function. The equation for the covariance matrix of the system state can now be derived, using the state equation:

$$\frac{d}{dt}\,(\overline{\underline{x}\,\underline{x}^T}) = \overline{\underline{x}\,\dot{\underline{x}}^T} + \overline{\dot{\underline{x}}\,\underline{x}^T} = \overline{\underline{x}\,\underline{x}^T}\,\underline{A}^T + \underline{A}\,\overline{\underline{x}\,\underline{x}^T} + \overline{u\underline{x}}\,\underline{c}^T + \underline{c}\,\overline{u\,\underline{x}^T} \tag{3.144}$$

Since the output of a linear system, excited by white noise, is a stationary process, $\overline{\underline{x}\,\underline{x}^T}$ is time invariant. Hence, its time derivative is equal to zero and the equation becomes:

$$\underline{\overline{X}}\,\underline{A}^T + \underline{A}\,\underline{\overline{X}} + \overline{u\underline{x}}\,\underline{c}^T + \underline{c}\,\overline{u\,\underline{x}^T} = \underline{0} \tag{3.145}$$

It is now necessary to determine the cross correlation between the input and the output. This is done by expressing the solution of the state equation in the well known form:

$$\underline{x}(t) = \underline{\Phi}(t,t_0)\,\underline{x}_0 + \int_0^{\infty} \underline{\Phi}(t,\tau)\,\underline{c}\,u(\tau)\,d\tau \tag{3.146}$$

where $\underline{\Phi}(t,t_0)$ is the state transition matrix satisfying the homogeneous state equation:

$$\dot{\underline{\Phi}}(t,t_0) = \underline{A}\,\underline{\Phi}(t,t_0);\ \underline{\Phi}(t_0,t_0) = \underline{I} \tag{3.147}$$

Multiplying Equation (3.146) by u(t) and taking the expected value on both sides gives the following result:

$$\overline{u(t)\,\underline{x}(t)} = \underline{\Phi}(t,t_0)\,\overline{u(t)\underline{x}_0} + \int_0^{t} \underline{\Phi}(t,\tau)\underline{c}\,\overline{u(t)u(\tau)}\,dt \tag{3.148}$$

The input and the initial condition vector are uncorrelated and only the steady-state output is considered such that the first term of this equation is zero. Equation (3.143) can then be substituted into the integral in which case:

$$\overline{u(t)\underline{x}(t)} = \int_0^t \underline{\Phi}(t,\tau)\underline{c}\ \omega\ \delta(t-\tau)\ d\tau = \frac{1}{2}\underline{c}\ \omega \tag{3.149}$$

This result, as well as its transpose, is then substituted into Equation (3.145) to give:

$$\underline{\overline{X}}\ \underline{A}^T + \underline{A}\ \underline{\overline{X}} = -\omega\ \underline{c}\ \underline{c}^T = -\omega\ \underline{C} \tag{3.150}$$

The value of the cost function on noise can, therefore, be determined by solving the same basic matrix equation as the one used to find the value of the quadratic performance index, except that the system matrix now includes the coefficients of the shaping filter as well as those of the system. The necessary conditions for a local minimum of this cost function can be obtained in the same manner as before and added to the previously derived equations in order to form the necessary conditions for the total performance index.

The addition of the noise cost function to the system performance index increases the amount of computations which must be performed in order to determine its minimum as was the case with multiple input/output systems. The numerical methods of Chapter 4 allow this task to be performed in a practical way although this will not be done in this report.

3.10 The Inverse Sensitivity Problem

The specification of component tolerances is an important part of any control system design effort. One method for determining these tolerances is to simulate the system dynamics and observe the effect of changes in the component parameters on the response. Although such a simulation is likely to be performed in the final stages of the design process it may be undesirable to do so in the early stages.

The method of inverse system sensitivity may be used in this case to make a quick estimation of these tolerances, especially when a performance index in the quadratic form is used in the design process

One approach to the inverse sensitivity problem would be to specify the maximum permissible value of some sensitivity index and find the corresponding component tolerance which, in general, would not result in a unique solution for multiple parameter variations. Such a sensitivity index was defined in Section 3.3 as the quantity, which represents the first order effect of parameter variations on the expected value of the performance index:

$$J_S = \int_0^\infty \overline{\delta \underline{x}^T \; \underline{Q} \; \delta \underline{x}} \, dt = \operatorname{tr}[\underline{Q} \; \overline{\delta \underline{X}}] \tag{3.151}$$

The matrix, $\overline{\delta \underline{X}}$, was then found by solving Equations (3.27), (3.34), (3.40) and (3.43), knowing the covariance of the parameter variation. Specifying the value of J_S does not, however, determine a unique value of $\overline{\delta \underline{X}}$. Even if this matrix were to be specified, the aforementioned equations are not very suitable for determining the corresponding covariances of the system coefficients, which are contained in the $\overline{\delta \underline{X}_0}$, $\underline{W}$ and $\underline{V}$ matrices of Section 3.3.

A much simpler approach can be taken, using the linearity of these equations. Thus, it is only necessary to compute the value of J_S for a single variation of a given parameter in order to determine J_S for all possible variations of this parameter. Assume, for instance, that J_S is computed for $\overline{\delta \xi_1^2}$, a specified mean square value of the variation of ξ. The value of J_S for any other mean square variation of this parameter is then given by:

$$J_{S_2} = J_{S_1} \left[\frac{\overline{\delta\xi_2^2}}{\overline{\delta\xi_1^2}} \right] \tag{3.152}$$

since Equations (3.27), (3.34), (3.40) and (3.43) are all linear in the solution matrices as well as the parameter covariance matrix, $\underline{R}$, which in this case contains only a single element, $\overline{\delta\xi^2}$. If the variations of any two parameters are uncorrelated, it can furthermore be shown that their contributions to the sensitivity index are additive, i.e. the total value of J_S is obtained by superimposing the effects of the independent sources. The contribution of correlated variations are then computed simultaneously.

These properties can be used to estimate the permissible range of the parameters under investigation. The relative effects of changes in the various independent parameters on the system performance can be determined by comparing the contributions of unit variations to J_S. This allows a quick trade-off between the parameter tolerances to be made. An estimate of the actual value of these tolerances can also be made, assuming that the maximum value of J_S can be specified. A method, which could possibly be used to determine this value of J_S, consists of computing the variation in the system response as one parameter is changed by one standard deviation, for instance. By computing the corresponding value of J_S a correlation between the deviations of the response and the sensitivity index can be established. The first order example of Section 3.7 will be used to illustrate the use of this method.

<u>Example 3.2</u>

Consider the system shown in Figure 3.4 with S_{OL} and z_1 as the free design parameters. The model response is the same as before and it is assumed that the nominal values of the design parameters have been chosen as:

$$S_{OL}^* = 3 \quad \text{and} \quad z_1^* = 1.5$$

These values cannot be changed, but the bounds of the variations of S_{OL} and z_1 are to be specified. The variations of these parameters are taken to be independent in which case J_S is given by Equation (3.137) as:

$$J_S = \frac{1}{36} \overline{\left(\frac{\delta S_{OL}}{S_{OL}^*}\right)^2} + \frac{1}{9} \overline{\left(\frac{\delta z_1}{z_1^*}\right)^2} \tag{3.153}$$

where the nominal values of the design parameters have been substituted. This expression shows that a specified mean square relative change in z_1 has four times as much influence on the expected value of the performance index as has the same change in S_{OL}. If no other information is available about the system components the relative tolerances of the parameters may be specified such that:

$$\overline{\left(\frac{\delta S_{OL}}{S_{OL}}\right)^2} = 4 \overline{\left(\frac{\delta z_1}{z_1}\right)^2}$$

with the objective of achieving a balanced design in the sense that an equally likely variation of either parameter have the same effect on the performance index.

A correlation between the value of J_S and the actual deviation of the response can be obtained from Figure 3.6 which shows the system response for $\pm$ 50% variation of the static sensitivity. Assuming, for the moment, that this is the standard deviation of a single variable parameter, namely S_{OL}, the value of J_S is found to be:

$$J_S = \frac{1}{144}$$

Using this value as the maximum value of J_S and the previously determined ratio of the variations, the desirable values of the mean square variations can be computed from Equation (3.153) as:

$$\overline{\left(\frac{\delta S_{OL}}{S_{OL}}\right)^2} = \frac{1}{8}$$

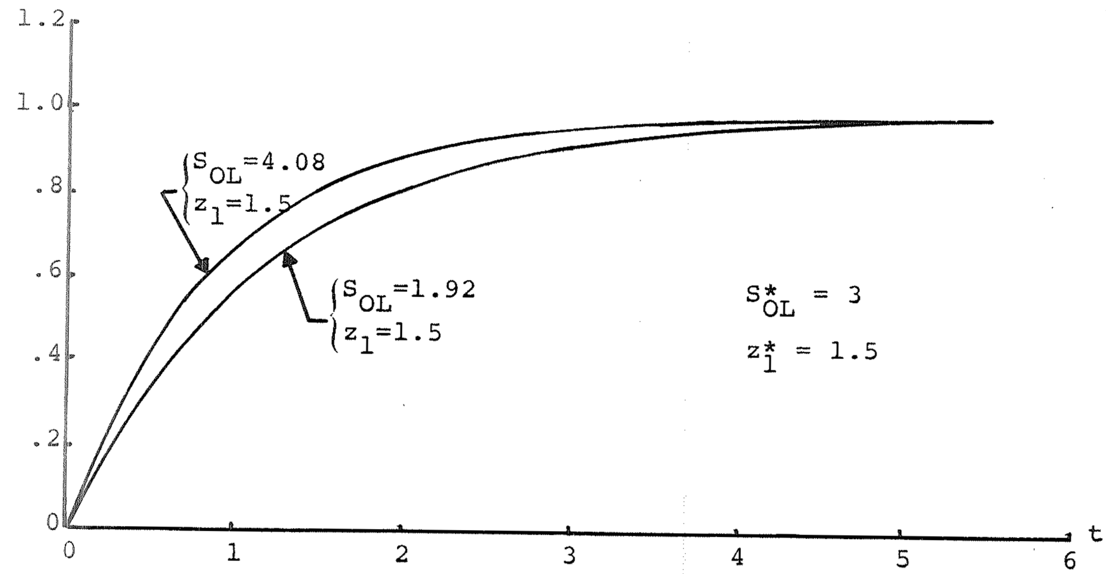
$$\overline{\left(\frac{\delta z_1}{z_1}\right)^2} = \frac{1}{32}$$

The tolerances may then be set at plus or minus one standard deviation, for instance, in which case:

$$\left|\frac{\delta S_{OL}}{S_{OL}}\right|_{max.} = \frac{1}{\sqrt{8}} \cong .36$$

$$\left|\frac{\delta z_1}{z_1}\right|_{max.} = \frac{1}{\sqrt{32}} \cong .18$$

The off-nominal responses of the system are shown in Figure 3.10 for the maximum allowable variations of each of the two parameters. The envelope of the output deviations due to variations of the open-loop gain is clearly very similar to the corresponding envelope for variations of the zero location. This is in agreement with the equal contribution of these variations to the sensitivity index. The magnitudes of the response deviations indicate whether the value of J_S, which was used to set the tolerances, was chosen too large or too small.



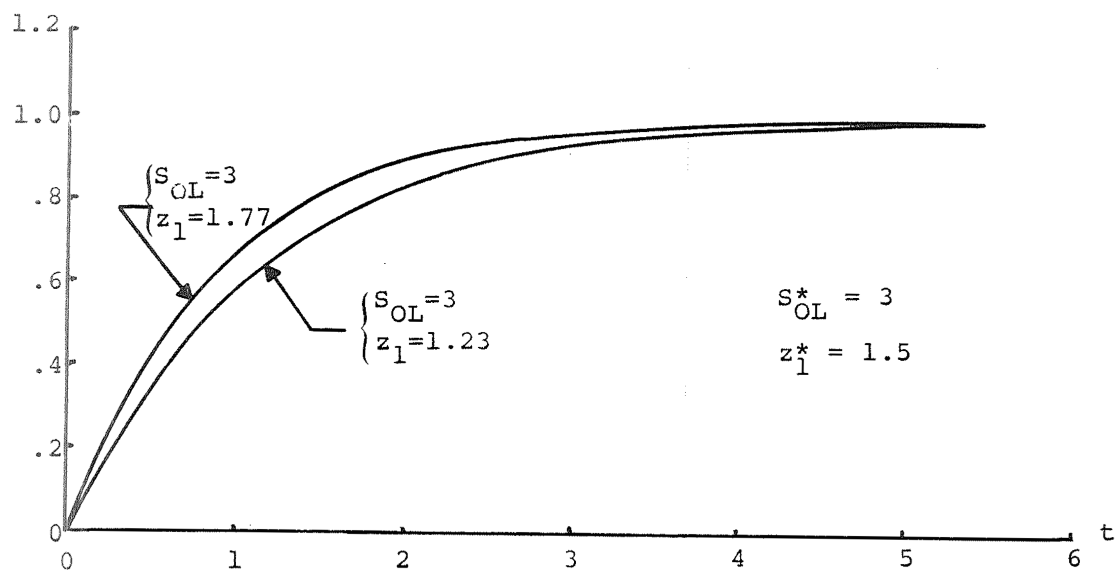


Figure 3.10 Off-nominal responses of first order system

3.11 Summary

The problem of designing a linear control system, which is subject to variations or uncertainties of some of its parameters, has been formulated as a constrained stochastic control problem. The configuration of the system is determined a priori by the designer with some free design parameters which can be chosen so as to optimize the system performance. Minimization of a quadratic performance index in terms of the transient system state vector is used for this purpose. Its value is random, however, due to parameter uncertainties, and the performance index is therefore defined as the expected value of the quadratic term, which is deterministic. This quantity was shown to be a sum of two terms, the first of which is simply the value of the performance index for the nominal parameter values. The second term depends only on the variations of the system parameters and can be used as an index of system sensitivity.

The necessary conditions for the minimum of this performance index could be obtained as straightforward matrix equations only because of the convenient form of the equations when the system matrix is in the phase-variable form. The problem of computing the numerical solution to these equations is left to the following chapter. Two specific forms of the performance index are considered, i.e. the model performance index and the integral square error index. The MPI is interpreted in a new way in terms of the error model of the system. This leads to a new and simpler method for dealing with systems with transfer zeros than was available before.

The integral square error index includes the model response explicitly as a function of time which makes it necessary to augment the necessary conditions. The effect,is, however, limited to the nominal part of the performance index and the sensitivity term is unaffected by the model response. An example, applying the ISE index to the design of a first order system, shows that using the expected value of this index is very useful in reducing the system sensitivity to variations in gain, given enough design freedom.

A method for including the effects of noise is considered whereby a new term expressing this effect is added to the system performance index. Finally, the problem of inverse sensitivity is discussed and a method developed for estimating the tolerances of statistically independent system parameters.

Chapter 4. Numerical Methods

4.1 Introduction

The necessary conditions for a local minimum of the expected value of a quadratic performance index were derived in Sections 3.3 and 3.4 of the previous chapter. These were found to consist of 2(n+2) matrix equations plus a single vector equation as shown by Equation (3.75), where n is the dimension of the system state. A solution of these equations must now be found in terms of the free design parameters. An analytic solution was obtained for a simple first order problem in Chapter 3, but this is impractical or impossible for any higher order problem especially if transfer function zeros are involved. It is therefore necessary to develop a numerical technique for obtaining the solution.

A well known method of this type is the gradient or steepest descent method, whereby the solution is found by iteratively moving in the negative direction of the local gradient vector in the parameter space. More specifically, the procedure consists of satisfying all the constraining equations of (3.75) except the last equation, which is a vector equation and becomes the expression for the local gradient at the current point in the parameter space as will be shown later.

Solutions to these constraining equations must be obtained for each step of the minimization process, using the current value of the design parameters. These solutions are then substituted into the last equation of Equation (3.75) in order to determine the gradient at this point. Thus, it is important that these equations be solved in a relatively efficient and accurate way, which is not a simple task for systems of high order. A problem of sixth order, for instance, requires the solution of 16 such equations with an equal number of solution matrices, each of which contains 36 elements. Thus, 576 scalar equations must be solved in this case although some of these are identical when the solution matrix is symmetric. One method which has been used in the past to solve equations of this type defines the solution as a steady-state solution of a matrix differential equation. For instance:

$$\dot{\underline{X}} = \underline{A}\,\underline{X} + \underline{X}\,\underline{A}^T + \underline{X}_0 \tag{4.1}$$

This equation is then integrated by some numerical procedure, such as the Runge-Kutta method, until a steady-state condition is reached, assuming that the system equations are stable. The initial condition for $\underline{X}$ is arbitrarily chosen, for instance, equal to the zero matrix.

This method is equivalent to solving for the time response of the system, which may require a great number of time steps before steady-state is reached. It has been found to be inefficient as well as inaccurate if the integration time step is not carefully selected, but has been used successfully in the solution of the deterministic design problem. In this case it is only necessary to solve two such matrix equations, whereas an additional 2(n+1) equations must be solved in order to determine the value and the gradient of the expected value of the performance index.

In following a suggestion by Professor Potter that the solution to Equation (4.1) could be simplified by writing the solution as a product of two matrices it was discovered that an explicit solution of Equation (4.1) can be obtained when the system matrix is in the phase-variable form. This solution leads to a very efficient method for determining the minimum of the performance index defined by Equation (3.13).

4.2 Solution of Matrix Equations

It was found in Section 3.3 of the previous chapter that the matrix equation (3.33) assumes a very convenient form when the system matrix is in the phase-variable form. This was seen by writing the equation column by column and obtaining an iterative relationship for any column vector of the solution matrix in terms of the adjacent column vector on its left hand side. This simplicity of form is a basic property of all the constraining matrix equations of Equation (3.75), and it can be used to develop a method for their solution. These equations are basically of four types although there is a great amount of similarity between them. The solution for each type will be developed separately.

4.2.1 $\underline{A}\,\underline{X} + \underline{X}\,\hat{\underline{A}}^T = \underline{C}$

This equation includes two system matrices: $\underline{A}$, which is an $n \times n$ system matrix, and $\hat{\underline{A}}$, which is an $\ell \times \ell$ system matrix. Both $\underline{A}$ and $\hat{\underline{A}}$ are assumed to be in the phase-variable form. $\hat{\underline{A}}$ is either identical to $\underline{A}$, in which case $\ell = n$, or it can be different from $\underline{A}$. An example of the latter case is the integral square error index, where $\hat{\underline{A}}$ is the model matrix and $\ell \leq n$. The unknown matrix, $\underline{X}$, and the constant matrix $\underline{C}$ must then be $n \times \ell$ matrices. The quantity $\underline{X}\,\hat{\underline{A}}^T$ can be written as:

$$\underline{X}\,\hat{\underline{A}}^T = \left[\begin{array}{c|c|c|c} \underline{x}_0 & \underline{x}_1 & \cdots & \underline{x}_{\ell-1} \end{array} \right] \begin{bmatrix} 0 & 0 & \cdots & 0 & -\alpha_0 \\ 1 & 0 & \cdots & 0 & -\alpha_1 \\ 0 & 1 & \cdots & 0 & -\alpha_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -\alpha_{\ell-1} \end{bmatrix} =$$

$$= \left[\begin{array}{c|c|c|c|c} \underline{x}_1 & \underline{x}_2 & \cdots & \underline{x}_{\ell-1} & -\alpha_0\underline{x}_0 - \alpha_1\underline{x}_1 - \cdots -\alpha_{\ell-1}\underline{x}_{\ell-1} \end{array} \right] \tag{4.2}$$

where $\underline{x}_i$ is the $(i+1)^{\underline{st}}$ column vector of $\underline{X}$. The effect of postmultiplying a general matrix by the transpose of a system matrix in the phase-variable form is to shift its columns to the left by one position. The last column is then replaced by a linear combination of all the column vectors, each of which is multiplied by the system coefficient of the same order. This property is useful in machine computations since the multiplication can be performed faster than in the case of two general matrices of the same dimension.

The matrix equation under consideration can now be written column by column using Equation (4.2):

$$\begin{aligned} \underline{A}\underline{x}_0 + \underline{x}_1 &= \underline{c}_0 \\ \vdots \qquad \vdots \quad &\quad \vdots \\ \underline{A}\underline{x}_{\ell-2} + \underline{x}_{\ell-1} &= \underline{c}_{\ell-2} \\ \underline{A}\underline{x}_{\ell-1} - \alpha_0\underline{x}_0 - \alpha_1\underline{x}_1 - \cdots - \alpha_{\ell-1}\underline{x}_{\ell-1} &= \underline{c}_{\ell-1} \end{aligned} \tag{4.3}$$

where $\underline{c}_i$ is the $(i+1)^{\underline{st}}$ column vector of the constant $\underline{C}$ matrix. Thus, an iterative relationship is obtained for the $i^{\underline{th}}$ column of $\underline{X}$ in terms of the $(i-1)^{\underline{st}}$ column. In order to start this process it is necessary to compute $\underline{x}_0$. An expression for $\underline{x}_0$ can be obtained by successively substituting for $\underline{x}_1 \ldots \underline{x}_{\ell-1}$ in the last equation of (4.3) using the first $\ell-1$ iterative expressions. This leads to the following equation in $\underline{x}_0$ alone:

$$\underline{E}_\ell \underline{x}_0 + \underline{E}_{\ell-1}\underline{c}_0 + \underline{E}_{\ell-2}\underline{c}_1 + \ldots + \underline{E}_1\underline{c}_{\ell-2} + \underline{E}_0\underline{c}_{\ell-1} = \underline{0} \tag{4.4}$$

where $\underline{E}_i$ is a matrix polynomial defined by:

$$\underline{E}_i = (-\underline{A})^i + \alpha_{i-1}(-\underline{A})^{i-1} + \ldots + \alpha_{\ell-i-1}(-\underline{A}) + \alpha_{\ell-i}\underline{I} \qquad 1 \leq i \leq \ell \tag{4.5}$$

and $\qquad \underline{E}_0 = \underline{I}$

$\underline{x}_0$ can now be computed by inverting $\underline{E}_\ell$:

$$\underline{x}_0 = -\underline{E}_\ell^{-1}\left[[\sum_{i=1}^{\ell} \underline{E}_{i-1}\ \underline{c}_{\ell-i}] \right] \tag{4.6}$$

It has been shown, that $\underline{E}_\ell$ can be inverted as long as $\underline{A}$ and $-\hat{\underline{A}}$ have no common eigenvalues [13], i.e. the system has no common poles with the adjoint system equation defined by:

$$\dot{\hat{\underline{x}}}(t) = -\hat{\underline{A}}^T\ \hat{\underline{x}}(t) \tag{4.7}$$

If $\underline{A}$ and $\hat{\underline{A}}$ are both system matrices of stable systems all their poles are in the left half complex plane. The adjoint system represented by Equation (4.7) has all its poles in the right half plane in this case, since the poles of a system and its adjoint form a mirror image about the imaginary axis. $\underline{E}_\ell$ can therefore be inverted as long as the condition of stability is satisfied. When $\hat{\underline{A}} = \underline{A}$, as is the case in some of the equations of (3.75), the same condition holds. It may also be observed that at the point of instability, i.e. as the poles cross the imaginary axis in the complex plane, the system poles coincide with the poles of its adjoint system and $\underline{E}_\ell$ is no longer invertible.

When $\underline{x}_0$ has been computed using Equation (4.6) the remaining column vectors of the $\underline{x}$ matrix are obtained using the iterative relationship:

$$\underline{x}_i = -\underline{A}\,\underline{x}_{i-1} + \underline{c}_{i-1} \qquad 1 \leq i < \ell \tag{4.8}$$

Finally, a very useful expression for computing the $\underline{E}_i$ matrices can be obtained from Equation (4.5):

$$\underline{E}_i = -\underline{A}\,\underline{E}_{i-1} + \alpha_{\ell-i}\,\underline{I} \qquad 1 \leq i \leq \ell \tag{4.9}$$

with

$$\underline{E}_0 = \underline{I}$$

4.2.2 $\underline{A}^T\underline{P} + \underline{P}\,\hat{\underline{A}} = \underline{C}$

The solution to this equation can be obtained in a similar way to the solution of the preceding matrix equation. The system matrices are the same as before and $\underline{C}$ is any constant $n\times\ell$ matrix. The product term $\underline{P}\,\hat{\underline{A}}$ can now be written as:

$$\underline{P}\,\hat{\underline{A}} = \left[\begin{array}{c|c|c|c} \underline{p}_0 & \underline{p}_1 & \cdots & \underline{p}_{\ell-1} \end{array}\right] \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & & -\alpha_{\ell-1} \end{bmatrix} =$$

$$= \left[-\alpha_0\underline{p}_{\ell-1} \;\middle|\; \underline{p}_0-\alpha_1\underline{p}_{\ell-1} \;\middle|\; \cdots \;\middle|\; \underline{p}_{\ell-2}-\alpha_{\ell-1}\underline{p}_{\ell-1} \right] \tag{4.10}$$

The effect of postmultiplying a general matrix by a system matrix in the phase-variable form is to shift its columns one position to the right, replacing the first column by zeros, in addition to subtracting a term of the from $\alpha_{i-1}\underline{p}_{\ell-1}$ from the $i^{\underline{th}}$ column of the new matrix. This is a useful relationship for machine computations. The matrix equation is then written out column by column as before, which gives:

$$\begin{aligned}
&\underline{A}^T\underline{p}_0 - \alpha_0\underline{p}_{\ell-1} = \underline{c}_0 \\
&\underline{A}^T\underline{p}_1 + \underline{p}_0 - \alpha_1\underline{p}_{\ell-1} = \underline{c}_1 \\
&\vdots \qquad \vdots \qquad \vdots \qquad \vdots \\
&\underline{A}^T\underline{p}_{\ell-1}+\underline{p}_{\ell-2} - \alpha_{\ell-1}\underline{p}_{\ell-1} = \underline{c}_{\ell-1}
\end{aligned} \tag{4.11}$$

These equations give an iterative relationship for the $i^{\underline{th}}$ column vector of $\underline{P}$ in terms of the $(i+1)^{\underline{st}}$ column vector as well as the last column vector. It is therefore necessary to start by computing the value of $\underline{p}_{\ell-1}$. This can be done by successive substitution for $\underline{p}_i$ in the $i^{\underline{th}}$ equation of (4.11), using the expression obtained for $\underline{p}_i$ from the $(i+1)^{\underline{st}}$ equation. Starting with the last equation, which gives $\underline{p}_{\ell-2}$ in terms of $\underline{p}_{\ell-1}$ this process is completed when all the unknown column vectors except $\underline{p}_{\ell-1}$ have been eliminated:

$$\underline{E}_\ell \, \underline{p}_{\ell-1} + (-\underline{A})^{\ell-1}\underline{c}_{\ell-1} + (-\underline{A})^{\ell-2}\underline{c}_{\ell-2} + \ldots + (-\underline{A})\underline{c}_1 + \underline{c}_0 = \underline{0} \qquad (4.12)$$

where $\underline{E}_\ell$ is the same matrix polynomial as defined by Equation (4.5). $\underline{p}_{\ell-1}$ is then obtained by inverting this matrix:

$$\underline{p}_{\ell-1} = -\underline{E}_\ell^{-1} \left[\sum_{i=0}^{\ell-1} (-\underline{A})^i \, \underline{c}_i \right] \qquad (4.13)$$

where the inverse of this matrix exists subject to the conditions outlined in the preceding subsection. The remaining column vectors are then easily computed by using the iterative relationship:

$$\underline{p}_i = \alpha_{i+1}\underline{p}_{\ell-1} - \underline{A}^T \underline{p}_{i+1} + \underline{c}_{i+1} \qquad 0 \le i \le \ell-2 \qquad (4.14)$$

The term expressed by the summation in Equation (4.13) is computed most conveniently by using the iterative equation:

$$\underline{d}_i = -\underline{A}^T \, \underline{c}_{\ell-i} + \underline{c}_{n-i-1} \qquad 1 \le i \le \ell-1 \qquad (4.15)$$

where

$$\underline{d}_0 = \underline{c}_{\ell-1}$$

and

$$\underline{d}_{\ell-1} = \sum_{i=0}^{\ell-1} (-\underline{A})^i \, \underline{c}_i \qquad (4.16)$$

4.2.3 Solution of $\underline{Z}_i$ and $\underline{\Lambda}_i$

The equations for the n×n $\underline{Z}_i$ matrices were derived in Section 3.3 as:

$$\underline{Z}_i + \underline{A} \, \underline{Z}_{i-1} + \underline{x}_0 \underline{v}_{i-1}^T = \underline{0} \qquad 0 < i \le n-1 \qquad (3.40)$$

$$\underline{A}\,\underline{Z}_{n-1} - a_{n-1}\underline{Z}_{n-1} - \cdots - a_0\underline{Z}_0 + \underline{x}_0\underline{v}_{n-1}^T = \underline{X}\,\underline{W} \qquad (3.43)$$

It is clear that $\underline{Z}_0$ must be determined first in order to compute the remaining matrices from Equation (3.40). This can be done by consecutive substitution into Equation (3.43) of the iterative expressions for $\underline{Z}_i$, starting with $\underline{Z}_{n-1}$. The following equation for $\underline{Z}_0$ is then obtained:

$$\underline{E}_n\underline{Z}_0 - \underline{E}_0\underline{x}_0\underline{v}_{n-1}^T - \underline{E}_1\underline{x}_0\underline{v}_{n-2}^T - \cdots - \underline{E}_{n-1}\underline{x}_0\underline{v}_0^T + \underline{X}\,\underline{W} = \underline{0} \qquad (4.17)$$

where $\underline{E}_i$ is the same matrix polynomial as expressed by Equation (4.5) with $\alpha_i = a_i$ and $\ell = n$:

$$\underline{E}_i = (-\underline{A})^i + a_{n-1}(-\underline{A})^{i-1} + \cdots + a_{n-i+1}(-\underline{A}) + a_{n-i}\underline{I} \qquad (4.18)$$

$$0 < i \leq n-1$$

$$\underline{E}_i = -\underline{A}\,\underline{E}_{i-1} + a_{n-1}\,\underline{I} \qquad (4.19)$$

and $\underline{E}_0 = \underline{I}$

$\underline{E}_n$ can always be inverted when $\underline{A}$ is the system matrix of a stable system. The solution for $\underline{Z}_0$ then becomes:

$$\underline{Z}_0 = \underline{E}_n^{-1}\left[\sum_{i=0}^{n-1} \underline{E}_i\underline{x}_0\underline{v}_{n-i-1}^T - \underline{X}\,\underline{W}\right] \qquad (4.20)$$

and the remaining $\underline{Z}_i$ matrices are computed from the iterative relationship:

$$\underline{Z}_i = -\underline{A}\,\underline{Z}_{i-1} - \underline{x}_0\underline{v}_{i-1}^T \qquad (4.21)$$

The equations for the Lagrangian matrices corresponding to the preceding equations for $\underline{Z}_i$ were derived in Section 3.4:

$$\underline{\Lambda}_1\underline{A} - a_0\underline{\Lambda}_n = 2\, p_{1n}\underline{I} \tag{3.58}$$

and

$$\underline{\Lambda}_{i-1} + \underline{\Lambda}_i\underline{A} - a_{i-1}\underline{\Lambda}_n - 2\, p_{in}\underline{I} = \underline{0} \qquad 1<i\leq n \tag{3.60}$$

where the p_{in} coefficients are members of the last column of the Lagrangian matrix $\underline{P}_2$, which is assumed to have a known solution at this point. Equation (3.60) is an iterative relationship which makes it possible to determine all the $\underline{\Lambda}_i$ matrices once $\underline{\Lambda}_n$ has been computed. The following equation for $\underline{\Lambda}_n$ is determined by successive substitution of Equation (3.60) into Equation (3.58) starting with i = 2:

$$\underline{\Lambda}_n\underline{E}_n = -2\left[p_{nn}(-A)^{n-1} + p_{(n-1)n}(-A)^{n-2} + \ldots + p_{2n}(-A) + p_{1n}\underline{I}\right] \tag{4.22}$$

where $\underline{E}_n$ is the same matrix polynomial as before and can therefore be inverted for a stable system to give:

$$\underline{\Lambda}_n = -2\left[\sum_{i=0}^{n-1} p_{(i-1)n}(-\underline{A})^i\right]\underline{E}_n^{-1} \tag{4.23}$$

The remaining matrices are then readily determined, since:

$$\underline{\Lambda}_{i-1} = a_{i-1}\underline{\Lambda}_n - \underline{\Lambda}_i\underline{A} + 2\, p_{in}\underline{I} \qquad 1<i\leq n \tag{4.24}$$

4.3 Numerical Accuracy

It has been shown that in order to obtain a numerical solution to the matrix equations considered in this section a great number of matrix operations must be performed, including a matrix inversion. Most of these involve multiplications by the system matrix $\underline{A}$, in some instances to $n^{\underline{th}}$ power. The speed and accuracy of the machine computations of these multiplications is enhanced by taking advantage of the phase-variable form of this matrix, as indicated in Sections 4.2.1 and 4.2.2 in the case of postmultiplication of a general matrix by $\underline{A}$ or its transpose. Analogous properties can be demonstrated for premultiplication by $\underline{A}$ of a general matrix, in which case the rows of this matrix get operated upon. For instance:

$$\underline{AB} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & & -a_1 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} \underline{b}_0^T \\ \hline \vdots \\ \hline \underline{b}_{n-1}^T \end{bmatrix} = \begin{bmatrix} \underline{b}_1^T \\ \hline \vdots \\ \hline \underline{b}_{n-1}^T \\ \hline -a_0\underline{b}_0^T - \cdots - a_{n-1}\underline{b}_{n-1}^T \end{bmatrix} \tag{4.25}$$

The effect here is to shift the rows of $\underline{B}$ up by one position and replace the last row by a linear combination of all the rows of $\underline{B}$. A similar expression can be given for the premultiplication by the transpose of the system matrix.

However, it has been found that the numerical accuracy of the preceding solutions is insufficient for most practical problems when the computations are performed with an eight decimal accuracy (single precision) using the standard Gauss-Jordan method for matrix inversions. One possible way of improving the accuracy consists of increasing the number of decimals to sixteen (double presision), which doubles the storage requirements in the computer memory. A more elegant method uses an iterative procedure for refining the solution until some desired accuracy has been achieved. This requires no additional storage space but may result in a longer computation time depending on the number of iterations performed. This technique will be described in detail for each type of matrix equation.

Consider first the equation of Section 4.2.1:

$$\underline{A}\ \underline{X} + \underline{X}\ \underline{\hat{A}}^T = \underline{C} \tag{4.26}$$

assuming that an initial solution has already been computed, using the appropriate equations. This solution, which is denoted by $\underline{\tilde{X}}$, is then substituted into the original equation, which gives:

$$\underline{A}\ \underline{\tilde{X}}_1 + \underline{\tilde{X}}_1\ \underline{\hat{A}}^T = \underline{\tilde{C}}_1 \tag{4.27}$$

where $\underline{\tilde{C}}_1$ is now the computed right hand side of the equation. By subtracting this equation from the original one the following result is obtained:

$$\underline{A}\Delta\underline{X}_1 + \Delta\underline{X}_1\underline{\hat{A}}^T = \Delta\underline{C}_1 \tag{4.28}$$

where

$$\Delta\underline{X}_1 = \underline{X} - \underline{\tilde{X}}_1 \tag{4.29}$$

and

$$\Delta\underline{C}_1 = \underline{C} - \underline{\tilde{C}}_1 \tag{4.30}$$

$\Delta\underline{X}$ is as yet unknown, but $\Delta\underline{C}_1$ is the difference between the computed and actual constant matrices and is therefore known. If this difference is zero it follows from Equation (4.28) that $\Delta\underline{X}$ is zero and the computed solution, $\underline{\tilde{X}}_1$, is exact. In most instances this will not be the case and Equation (4.28) may be solved in order to determine $\Delta\underline{X}$.

This is done by simply replacing the $\underline{C}$ matrix by $\Delta\underline{C}_1$. A new computed solution is then obtained by adding the computed value of $\Delta\underline{X}$, which is denoted by $\Delta\tilde{\underline{X}}_1$, to the previously computed solution matrix:

$$\tilde{\underline{X}}_2 = \tilde{\underline{X}}_1 + \Delta\tilde{\underline{X}}_1 \tag{4.31}$$

The process may then be repeated as often as required to achieve the desired degree of accuracy, assuming that convergence is experienced. The $i^{\underline{th}}$ computed solution is then given by:

$$\tilde{\underline{X}}_i = \tilde{\underline{X}}_1 + \Delta\tilde{\underline{X}}_1 +\ldots+ \Delta\tilde{\underline{X}}_{i-1} \tag{4.32}$$

The computed $\underline{C}$ matrix is similarly obtained by adding the contribution of each iteration to the previously computed right hand side of Equation (4.26):

$$\tilde{C}_i = \tilde{C}_1 + \Delta\tilde{C}_1 +\ldots+ \Delta\tilde{C}_{i-1} \tag{4.33}$$

It should be noted that once the coefficient polynomials $\underline{E}_i$ and $\underline{d}_{\ell-1}$ have been computed for the first solution, it is unnecessary to recompute them for each iteration, which is a matter of some practical importance. There is, of course, a trade-off here between storage space and computater time, since a total of n (n×n) matrices must be stored in order to avoid recomputing them. It should also be pointed out that no effort has been made to improve the accuracy of the inversion of the $\underline{E}_\ell$ matrix, which is an area of potential improvement although the degree of accuracy of this operation has not been determined.

The iteration procedure, which has just been described, has been found to have very satisfactory performance both with respect to accuracy and rate of convergence. The accuracy is determined by comparing the specified constant matrix, $\underline{C}$, with the corresponding computed matrix, which is obtained from Equation (4.33) for the $i^{\underline{th}}$ iteration. The maximum percentage error of all the elements of $\tilde{\underline{C}}_i$ can then be used as an index of convergence. Typically, the rate of improvement of this error has been found to be about at least two orders of magnitude per iteration and often more. The maximum permissible value of this error has been arbitrarily specified as one part per 10^{10} in the numerical examples of this thesis, using double precision in some of the critical matrix operations.

The same basic method can be used to refine the solution of the remaining matrix equations although some explanations are in order regarding the computation of the equations for $\underline{Z}_i$ and $\underline{\Lambda}_i$. Assuming that a first solution has been computed for all the $\underline{Z}_i$ matrices, for instance, these can be substituted into Equation (3.43) to give:

$$\underline{A}\tilde{\underline{Z}}_{n-1}^{(1)} - a_{n-1}\tilde{\underline{Z}}_{n-1}^{(1)} - \ldots - a_0\tilde{\underline{Z}}_0^{(1)} + \underline{x}_0\underline{v}_{n-1}^T = \underline{X}\,\tilde{\underline{W}}_1 \tag{4.34}$$

where $\tilde{\underline{Z}}_i^{(1)}$ denotes the first computed value of $\underline{Z}_i$ and $\underline{X}\,\tilde{\underline{W}}_1$ is the computed value of $\underline{X}\,\underline{W}$. Subtracting this equation from the original equation then gives an equation for the correction:

$$\underline{A}\Delta\underline{Z}_{n-1}^{(1)} - a_{n-1}\Delta\underline{Z}_{n-1}^{(1)} - \ldots - a_0\Delta\underline{Z}_0^{(1)} = \Delta(\underline{X}\,\underline{W}) \tag{4.35}$$

where

$$\Delta\underline{Z}_i^{(1)} = \underline{Z}_i - \tilde{\underline{Z}}_i^{(1)} \tag{4.36}$$

and

$$\Delta(\underline{X}\,\underline{W}) = \underline{X}\,\underline{W} - \tilde{\underline{X}}\,\underline{W}_1 \tag{4.37}$$

The computed solutions, $\underline{\tilde{Z}}_i^{(1)}$, can also be substituted into the iterative relationship of Equation (3.40) to give:

$$\underline{\tilde{Z}}_i^{(1)} + \underline{A}\,\underline{\tilde{Z}}_{i-1}^{(1)} + \underline{x}_0\underline{v}_{i-1}^T = \underline{0} \qquad 0<i\leq n-1 \tag{4.38}$$

The question may be raised here why the right hand side of this equation is equal to zero considering the fact that the matrices $\underline{\tilde{Z}}_i^{(1)}$ and $\underline{\tilde{Z}}_{i-1}^{(1)}$ are inexact solutions in general. The answer to this question is that Equation (4.38) is correct within the limitation imposed by round-off errors, since $\underline{\tilde{Z}}_i^{(1)}$ was obtained by computing the right hand side of the following equation:

$$\underline{\tilde{Z}}_i^{(1)} = -\underline{A}\,\underline{\tilde{Z}}_{i-1}^{(1)} - \underline{x}_0\underline{v}_{i-1}^T \tag{4.39}$$

As a consequence it is not necessary to actually substitute the solutions into Equation (4.38), but it can be subtracted from the original equation to yield an iterative equation for $\Delta\underline{Z}_i^{(1)}$:

$$\Delta\underline{Z}_i^{(1)} + \underline{A}\Delta\,\underline{Z}_{i-1}^{(1)} = \underline{0} \qquad 0<i\leq n-1 \tag{4.40}$$

The solutions to Equations (4.35) and (4.40) can then be computed and added to the previous solution:

$$\underline{\tilde{Z}}_i^{(2)} = \underline{\tilde{Z}}_i^{(1)} + \Delta\underline{\tilde{Z}}_i^{(1)} \tag{4.41}$$

This process is then repeated until the desired accuracy is achieved. It is interesting that the n computed matrix solutions only have to be substituted into one of the n matrix equations in order to permit the computation of the corrections. Equations (3.58) and (3.60) for $\underline{\Lambda}_i$ have a similar property, whereby the accuracy of the solutions can be determined by substitution into the first of these equations. For instance:

$$\underline{\tilde{\Lambda}}_1^{(1)} \; \underline{A} - a_0\underline{\tilde{\Lambda}}_n^{(1)} = \underline{\tilde{M}}_1 \tag{4.42}$$

where $\underline{\tilde{\Lambda}}_i^{(1)}$ is the first computed value of $\underline{\Lambda}_i$ and $\underline{\tilde{M}}_1$ is the resulting right hand side. Substitution of the solution matrices into the iterative equations (3.60) gives no additional information about the numerical accuracy of the solution, since these equations are always satisfied within the round-off error limitations. The fact that only $\underline{\tilde{\Lambda}}_1$ and $\underline{\tilde{\Lambda}}_n$ are needed for the accuracy test has some practical implications with respect to storage requirements, since it is not necessary to store the intermediate solutions for $\underline{\tilde{\Lambda}}_i$ as will be seen later. The equations for the corrections to $\underline{\tilde{\Lambda}}_i^{(1)}$ are obtained in a way analogous to the development of Equation (4.40).

This method for refining the solutions for $\underline{Z}_i$ and $\underline{\Lambda}_i$ has been found to have accuracy and rate of convergence which is very similar to that described above for Equation (4.26). Computer programs for performing the numerical solutions of these equations are presented in Appendix C of this document.

4.4 Computation of the Gradient

It was mentioned in the introduction of this chapter that the gradient of the expected value of the quadratic performance index is expressed by the last equation of Equation (3.75) when all the remaining equations of the necessary conditions are satisfied. Thus, the gradient is given by:

$$\underline{g} = \frac{\partial \tilde{J}}{\partial \underline{p}} = -\left[\frac{\partial \underline{a}}{\partial \underline{p}}\right]^T \left[\underline{X}(\underline{P}_1+\underline{P}_1^T) \;\; +2\overline{\delta \underline{X}}\; \underline{P}_2 + \sum_{i=1}^{n} \underline{Z}_{i-1}\underline{\Lambda}_i\right] \underline{n}$$

$$+\left[\frac{\partial \underline{a}}{\partial \underline{p}}\right]^T \left[\underline{P}_1+\underline{P}_1^T\right] \underline{x}_0 + \left[\sum_{i=1}^{n} \left[\frac{\partial \underline{v}_{i-1}}{\partial \underline{p}}\right]^T \underline{\Lambda}_i\right] \underline{x}_0 + \left[\frac{\partial \underline{x}_0}{\partial \underline{p}}\right]^T \left[\sum_{i=1}^{n} \underline{\Lambda}_i^T \; \underline{v}_{i-1}\right]$$

$$- \sum_{i=1}^{n} \left[\mathrm{tr}\left[\underline{\Lambda}_n \underline{Z}_{i-1}\right]\left[\frac{\partial a_{i-1}}{\partial \underline{p}}\right]\right] + \underline{e} - \underline{f} \tag{4.43}$$

where the elements of the $\underline{e}$ and $\underline{f}$ vectors are defined by:

$$e_i = 2\;\mathrm{tr}\left[\underline{P}_2\left[\frac{\partial^2 \underline{x}_0}{\partial p_i \partial \underline{\xi}}\right] \;\underline{R}\left[\frac{\partial \underline{x}_0}{\partial \underline{\xi}}\right]^T\right] \tag{4.44}$$

$$f_i = \mathrm{tr}\left[\left[\underline{\Lambda}_n \underline{X} + \underline{X}\;\underline{\Lambda}_n^T\right]\left[\frac{\partial^2 \underline{a}}{\partial p_i \partial \underline{\xi}}\right] \underline{R}\left[\frac{\partial \underline{a}}{\partial \underline{\xi}}\right]^T\right] \tag{4.45}$$

and

$$\underline{n}^T = [0,0, \;\ldots\;, 1]$$

This can be verified by considering the first order variation of $\underline{J}$ with respect to $\underline{p}$, which is expressed by Equation (3.74) when all other terms of $\delta\tilde{J}$ are set equal to zero.

The gradient expression given by Equation (4.43) must be evaluated at any specified point in the free design parameter space in order to determine the direction in this space, which leads to a smaller value of the performance index. (It should be noted that the variable parameters all assume their nominal values although the notation indicating this has been dropped for simplification.) For this purpose it is necessary to compute the solutions to the constraining equations, which must be satisfied for the current values of the free design parameters. These are rewritten here for completeness:

$$\underline{A}\,\underline{X} + \underline{X}\,\underline{A}^T = -\underline{X}_0$$

$$\underline{P}_2\underline{A} + \underline{A}^T\underline{P}_2 = -\underline{Q}$$

$$\underline{Z}_i + \underline{A}\,\underline{Z}_{i-1} = -\underline{x}_0\underline{v}_{i-1}^T \qquad 0<i\leq n-1$$

$$\underline{A}\,\underline{Z}_{n-1} - \sum_{i=0}^{n-1} a_i\underline{Z}_i = \underline{X}\,\underline{W} - \underline{x}_0\underline{v}_{n-1}^T$$

$$\underline{\Lambda}_{i-1} + \underline{\Lambda}_i\underline{A} - a_{i-1}\underline{\Lambda}_n - 2\,p_{in}\,\underline{I} = \underline{0} \qquad 1<i\leq n$$

$$\underline{\Lambda}_1\underline{A} - a_0\underline{\Lambda}_n - 2\,p_{1n}\,\underline{I} = \underline{0}$$

$$\underline{A}\,\overline{\delta X} + \overline{\delta X}\,\underline{A}^T + \underline{U} = -\overline{\delta X}_0$$

$$\underline{P}_1\underline{A} + \underline{A}^T\underline{P}_1 - \underline{W}\,\underline{\Lambda}_n = -\underline{Q} \qquad (4.46)$$

where $\underline{U}$ is the function of the traces of the $\underline{Z}_i$ matrices as given by Equation (3.46) and p_{in} is a component of the last column of the $\underline{P}_2$ matrix. The equations have been rearranged such that all the matrices on their right hand sides are known functions of the design parameters, $\underline{p}$, which were defined in Sections 3.2 and 3.3. These matrices as well as the coefficients of the equations must all be computed for the current value of $\underline{p}$ before the solutions can be obtained.

This task is relatively straightforward, given the definition of all the terms, but not necessarily simple, since it requires the computation of all the coefficients of the closed-loop transfer function and system initial conditions as well as the derivatives of these terms with respect to the free design parameters, $\underline{p}$, and the variable parameters, $\underline{\xi}$. These derivatives are also required for the computation of the gradient and a numerical method for determining them is described in Appendix B. It is of course possible to determine analytical expressions for these derivatives, but in all but the simplest problems this would be a tedious task or the form, in which the design parameters enter into the system equations, would have to be restricted. By computing numerical approximations to the derivatives the method used to find the closed-loop system coefficients is unimportant and the design parameters may be entered into the system equations in any desirable way.

The following table references the equations defining some of the terms which must be computed before a solution to the gradient expression can be obtained.

term	equation
$\underline{x}_0$	(3.11)
$\underline{X}_0$	(3.27)
$\overline{\delta\underline{X}_0}$	(3.35)
$\underline{V}$	(3.42)
$\underline{W}$	(3.44)
$\dfrac{\partial \underline{v}_i}{\partial \underline{p}_j}$	(3.69)

Table 4.1 References of definitions

The remaining derivative terms are self-explanatory.

Once all the coefficients and constants have been determined, the equations for $\underline{X}$, $\underline{P}_2$, $\underline{Z}_i$, $\underline{\Lambda}_i$, $\overline{\delta \underline{X}}$, and $\underline{P}_1$ are solved using the techniques of Section 4.2, which were found to be particularly suitable for machine computations. The equations must be solved in the same order as they appear in Equation (3.46), since only the first two of these are independent of the remaining equations each of which depends on the solution of an equation above it. The solutions are then substituted into Equation (4.43) in order to determine the value of the gradient. The value of the performance index is also easily obtained by using Equation (3.24), which expresses $\overline{J}$ as:

$$\overline{J} = \mathrm{tr}\left[\underline{Q}\;(\underline{X} + \overline{\delta \underline{X}})\right] \tag{3.24}$$

The gradient expression for the integral square error performance index contains two terms, representing the explicit effect of the model, in addition to the quantities of Equation (4.43). These terms are obtained from Equation (3.121) and the gradient of the ISE index can be written as:

$$\frac{\partial \overline{J}_{ISE}}{\partial \underline{p}} = \frac{\partial \overline{J}_{QPI}}{\partial \underline{p}} - \left[\frac{\partial \underline{a}}{\partial \underline{p}}\right]^T \hat{\underline{Y}}\, \underline{P}_3\, \underline{\eta} + \left[\frac{\partial \underline{x}_0}{\partial \underline{p}}\right]^T \underline{P}_3^T\, \hat{\underline{x}}_0 \tag{3.47}$$

where QPI refers to the quadratic performance index. $\hat{\underline{Y}}$ and $\underline{P}_3$ are the solutions of the constraining equations (3.118) and (3.120) which must be solved in addition to the equations (4.46). The value of the ISE index is evaluated by the following equation:

$$\overline{J}_{ISE} = \overline{J}_{QPI} + \mathrm{tr}\left[\underline{Q}\left[\hat{\underline{X}} - 2\hat{\underline{Y}}\right]\right] \tag{3.48}$$

where $\hat{\underline{X}}$ is the solution of Equation (3.117), which is not a function of the system design parameters and needs only be solved once for any specified model. All these equations are of the type discussed in Section 4.2.

A few remarks of practical interest can be made about the calculation of the gradient expression. Referring back to Equation (4.43) it is noted that the first term is postmultiplied by a vector whose only non-zero element is the last component. As a consequence, it is only necessary to compute the last column of the matrix products of this term, which reduces the number of computations required. It is also of considerable practical importance that the $\underline{\Lambda}_i$ matrices do not have to be stored for all values of i, because of the convenient way in which they enter into the gradient expression. The contribution of these matrices to the gradient can then be updated iteratively as corrections to the solutions are computed. The savings in storage space can be considerable for high order systems, since n^3 elements are involved. All the $\underline{Z}_i$ matrices must be stored, however, since these matrices are solved forward, starting with $i=0$, whereas the $\underline{\Lambda}_i$ matrices are solved backwards, which makes it impossible to compute the corresponding product terms in Equation (4.43), unless either $\underline{Z}_i$ or $\underline{\Lambda}_i$ are stored for all values of i.

4.5 Minimization Algorithm

The gradient expression of Section 4.3 can be used in a number of procedures, which determine the minimum of the corresponding function, $\bar{J}$, with respect to the specified variables, i.e. the free design parameters in this case. The simplest of these is the steepest descent method whereby the values of these parameters are incremented in the direction of the negative gradient vector in order to achieve a reduction in the value of $\bar{J}$. The starting values of the parameters must be specified as well as rules for controlling the magnitude or step size of the increments in the parameter space. This is basically the technique, which has been used in solving the design examples of this thesis. Thus, the change in the free design parameters at each point is given by:

$$\delta \underline{p} = -s \frac{\underline{g}}{|\underline{g}|} = -s\underline{h} \tag{4.49}$$

where s is the magnitude of the step size and $\underline{g}$ is the gradient of $\overline{J}$. This change in $\underline{p}$ will always reduce the value of $\overline{J}$, assuming that $\underline{g}$ is not the null vector and the step size is small enough, such that the change in $\overline{J}$ is approximately first order. The step size must be controlled very carefully in order to insure reasonable progress towards the minimum of $\overline{J}$ without invalidating the first order approximation.

The parabolic approximation has been found to be very useful for this prupose. The change in $\overline{J}$ is then approximated to second order in the direction of the gradient as:

$$\Delta\overline{J} \cong \underline{g}^T \delta\underline{p} + \frac{1}{2} \delta\underline{p}^T \underline{G} \, \delta\underline{p} \tag{4.50}$$

where $\underline{G}$ is the second derivative of $\overline{J}$ with respect to $\underline{p}$:

$$\underline{G} = \frac{\partial^2 \overline{J}}{\partial \underline{p}^2} \tag{4.51}$$

By substituting for $\delta\underline{p}$ from Equation (4.49) $\Delta\overline{J}$ becomes:

$$\Delta\overline{J} \cong -s \, \underline{g}^T \underline{h} + \frac{1}{2} s^2 \underline{h}^T \underline{G} \, \underline{h} \tag{4.52}$$

The minimum value of $\Delta\overline{J}$ with respect to s can be determined by differentiation and is obtained when:

$$s = \frac{\underline{g}^T \underline{h}}{\underline{h}^T \ \underline{G} \ \underline{h}^T} \tag{4.53}$$

This value of s determines the step size in the direction of the negative gradient from the current point in parameter space to the minimum of $\bar{J}$ as expressed by the second order approximation. The denominator of Equation (4.53) is, however, unknown but can be computed at any given point at which the value of $\bar{J}$ and its gradient are known, if the value of $\bar{J}$ is also known at another point along the direction of $\underline{h}$. Substituting these values into Equation (4.50) and rearranging gives:

$$\underline{h}_1^T \ \underline{G}_1 \ \underline{h}_1 \cong \frac{2}{s_1^2} \left(\bar{J}_2 - \bar{J}_1 + s_1 \frac{\underline{g}_1^T \ \underline{g}}{|\underline{g}|} \right)$$

where $\bar{J}_1$ and $\bar{J}_2$ are the values of $\bar{J}$ at the two points and s_1 is the distance between them. This expression must be positive in order for $\Delta\bar{J}$ to have a minimum along the direction of $\underline{h}$.

The step size which gives the distance from point 1 to the minimum value is then obtained from Equation (4.53) by substituting the computed value of $\underline{h}^T \ \underline{G} \ \underline{h}$ at this point. Basically two conditions are specified, which cause a parabolic step to be taken:

- the change in the performance index is positive and the step size must be reduced
- the scalar product of the gradient vectors at two consecutive points in parameter space is negative.

The first of these needs no explanation since it indicates that the minimum of $\bar{J}$ in the direction of $-\underline{g}$ has been overstepped. It is clear, however, that this minimum can be passed even though the value of $\bar{J}$ does not increase as, for instance, in the well known ravine problem where the process steps back and forth across a valley in the function space, making very small progress towards the minimum. The second condition is introduced in order to alleviate this problem by taking a parabolic step whenever the gradient at a given point has a negative projection on the previous gradient. This means that, in three-dimensional parameter space, the parabolic step is used when the gradient turns through more than 90° from one point to the next. This effectively prevents the straddling motion by locating the minimum of a ravine, when this type of behaviour is detected.

The amount by which the step size can be modified by the parabolic approximation has been arbitrarily limited, such that:

$$0.1\ s_i \leq s_p \leq 1.5\ s_i$$

where s_p and s_i are the parabolic and regular gradient steps, respectively. The step size can, therefore, be increased as well as decreased, which is of advantage in some instances. The following means of step size control are also included:

- step size is doubled if the difference between the actual and predicted changes in $\bar{J}$ is within a specified percentage value
- step size is halved if the computed curvature of $\bar{J}$ is found to be negative when attempting a parabolic step.

The first of these is used to determine the validity of the linear approximation. Thus, if the step size is well within the linear range it is likely that more improvement could be achieved in the value of $\overline{J}$ if a bigger step size were used. The second condition indicates a breakdown of the parabolic approximation and the step size is arbitrarily cut in half which often results in a more accurate value of the curvature calculation. The minimization process is terminated when either of the following conditions are satisfied:

- the parabolic step fails to make progress in three consecutive attempts
- both the actual and predicted computed improvements in $\overline{J}$ are smaller than a specified value

There has been no attempt to optimize the rate of convergence here but the procedure has been found to be reliable although convergence is relatively slow in the vicinity of the minimum, as is the case with most simple gradient techniques.

Chapter 5. Application to Flight Control Systems

5.1 Introduction

A flight vehicle is typically operated over a wide range of flight conditions with associated changes in its dynamic characteristics. Furthermore, these characteristics are not always accurately known for any given flight condition, especially before the vehicle has been flight tested.

Consequently, a flight control system may often be required to achieve some desirable performance despite uncertainties or specified changes in the vehicle's dynamics. Thus, the design of flight control systems is a logical area for the application of any method which takes such changes and uncertainties into account. Before applying the method of Chapter 3 to specific examples, it is appropriate to develop a general approach to problems of this type.

5.2 A Sensitivity Design Procedure

The sensitivity design method of Chapter 3 was developed on the basis of first order variations about a nominal time response of the system. Its application is, therefore, likely to be most useful when the trend in the system response, as these parameters are varied, can be approximated by first order effects. This does not necessarily mean that the changes in the parameters have to be small. It was seen in the example of Section 3.10, for instance, that the total range of the output response deviations of a first order system due to $\pm 50\%$ changes in static sensitivity was well predicted by a linear approximation, although the deviations were not symmetric about the nominal response. The sensitivity index, based on the linear deviations of the response, was also found to a useful indicator of the effect to these changes on the system response.

It was shown in Section 3.3 that the expected value of the quadratic performance can be separated into two parts, which can be written:

$$\overline{J} = J_* + J_s \tag{5.1}$$

where J_* is the value of the performance index when all the design parameters take on their nominal values. J_s expresses the effects of uncertainties in the variable design parameters on $\overline{J}$ and is referred to as an index of system sensitivity.

The basic approach to the sensitivity design, using the method developed in this thesis, can now be stated in terms of the following steps:

1) the configuration of the control system is chosen in an attempt to satisfy the specifications on nominal system response
2) the free design parameters are optimized by determining the minimum of the nominal value of a suitable quadratic performance index.

The choice of the feedback variables and the required compensation is mainly determined by the desire to obtain good nominal system performance at this stage. It is reasonable, however, to give some consideration to the influence of the configuration on system sensitivity. Some of the methods reviewed in Section 2.6 may be useful for this purpose.

The value of the sensitivity index can be computed at this point for each of the variable design parameters in order to indicate the relative importance of each variation on the system performance. If the design obtained by these steps meets all the specifications for all specified values of the variable design parameters, there is no need to go any further, since a satisfactory design has been found. If, on the other hand, the design is satisfactory for nominal values of the variable design parameters, but is unacceptable for the expected variations of these parameters, the following step is performed:

3) the expected value of the performance index is minimized for a specified value of the covariance matrix of the variable parameters.

If the system is still too sensitive, the effects of the uncertainties on the expected value of the performance index can be increased and the minimization repeated. This can be done by multiplying the covariance matrix of the parameter variations by a constant factor, which scales these variations without changing their relative relationship. The result is that more emphasis is placed on reducing the sensitivity index than before. If the design is still not satisfactory, it is concluded that the configuration chosen does not have the capabilities to meet the system specifications under the stated conditions of parameter uncertainty. A new configuration must then be chosen and the process repeated.

5.2.1 Trade-off Parameter, μ

For a fixed configuration which has a limited number of free design parameters, the improvement in the expected value of the performance index during step number 3 is usually obtained in such a way that the nominal value, J_*, is increased as the sensitivity index, J_s, is decreased. This means that the nominal system performance, as measured by the nominal index J_*, deteriorates somewhat, which is the price that must be paid for lower sensitivity. The sum of the changes in J_* and J_s must be negative, however, as long as the value of $\bar{J}$ is reduced. This is so, because:

$$\Delta\bar{J} = \Delta J_* + \Delta J_s < 0 \tag{5.2}$$

Thus, the improvement in the sensitivity index, J_s, is always greater than the corresponding deterioriation of the nominal value of the performance index.

The ability of a given system configuration to reduce the sensitivity of the nominal design can be judged on the basis of the ratio:

$$\mu = \frac{|\Delta J_s| - |\Delta J_*|}{|\Delta J_s|} \tag{5.3}$$

which lies in the range

$$0 \leq \mu \leq 1$$

where ΔJ_s and ΔJ_* represent the effect of minimizing the expected value of the performance index as compared with the values of J_s and J_* corresponding to the minimum of J_*. When the minima of J_* and $\bar{J}$ coincide, the value of J_s cannot be reduced any further by minimization of $\bar{J}$. Furthermore, the ratio of Equation (5.3) has the limit of zero in this case, as can be shown by considering a first order change in $\bar{J}$ due to variations of the free design parameters. To first order, this change must be zero at the minimum of $\bar{J}$, such that:

$$\tilde{\delta}\bar{J} = \tilde{\delta}J_* + \tilde{\delta}J_s = 0 \tag{5.4}$$

This can be used to show that:

$$\lim_{\Delta\bar{J}\to 0} \mu = 1 - \lim_{\tilde{\delta}\bar{J}\to 0} \frac{|\delta J_*|}{|\delta J_s|} = 0 \tag{5.5}$$

If the value of $\bar{J}$ can be minimized without affecting the minimum value of J_*, it is possible to obtain an improvement in sensitivity without impairing the nominal performance of the system as expressed by J_*. In this case, μ takes on a value of unity, since $\Delta J_* = 0$. This can only occur if the minimum value of J_* with respect to the free design parameters is not a unique function of these parameters. In addition, the minima of J_* and J_s would have to coincide at some point in the parameter space, which is an unlikely occurence.

Most designs fall somewhere in between these extremes. Thus, the higher the value obtained for μ, the more improvement can be achieved in system sensitivity for a given change in the nominal performance.

It should be emphasized that the final design can only be judged on the basis of how well it satisfies the original system specifications. The minimization of the performance index, or its expected value, is only a means to that end but does not guarantee an acceptable design by itself.

The values of these indices can, however, be used to give an estimate of the relative merits of different designs of the same system. This is indicated by the fact that doubling the deviation of the system response has the effect of quadrupling the value of the sensitivity index. The relative change of the sensitivity index may, therefore, be used to estimate the corresponding change in the sensitivity of the system response.

5.3 Flight Control Systems

Flight control systems are often separated into two main categories of stability augmentation systems and automatic guidance systems.

The stability augmentation systems are used to alter the basic dynamics of the vehicle so that it may be controlled by a human pilot with relative ease. The specifications for these systems are set by the handling qualities requirements for the various vehicles, which may result in a wide range of acceptable designs. These requirements are often expressed in terms of the desirable locations of the dominant system mode singularities.

The automatic guidance systems, in addition to providing system stability, are required to be compatible with guidance commands, which determine the system response specifications. These specifications are often stated in terms of the time response of the system to a standard input signal and may put a severe demand on the control system. Thus it may be very difficult to meet the requirements for a fast and well damped response to these input commands under the conditions of uncertainty or changes in the vehicle characteristics. This task may be made even harder by the existence of lightly damped system modes, such as structural bending modes, which may become destabilized in the attempt to satisfy the specifications on the dominant modes.

Variations in the operating environment represents one of the most common sources of change in the dynamics of a flight vehicle. These changes are often very large, and cannot be considered on the basis of linear perturbations from a single flight condition. In many cases it is sufficient, however, to consider only a limited number of representative flight conditions in order to insure that the performance specifications are met throughout the flight envelope. Typically, these performance requirements are not the same in all flight regimes of the vehicle. It is, therefore, unlikely that a single design, with all the free design parameters set constant, will be desirable for controlling the vehicle in widely different flight conditions, although this approach has been shown to work in some cases [19].

More commonly, some type of adaptation is likely to be used as the flight conditions change from one regime to another. Thus, for instance, some gains may be varied continuously or in increments as a function of specified flight variables, and elements of compensation may be engaged or disengaged depending on the operating condition. Despite adjustments of this type, it is desirable that the system be inherently insensitive to small changes in the flight conditions. Closed-loop adaptation, using a reference model or parameter identification, would probably be employed only when simpler techniques fail to produce a satisfactory design.

The dynamic characteristics of the flight vehicle are usually described by linear perturbation equations about the equilibrium flight conditions, which represent the operating environment of the vehicle. The coefficients of these equations, i.e. the stability derivatives, must be determined analytically or by experimental tests. These tests may be made with representative models of the vehicle under simulated conditions or by using the vehicle itself under actual conditions, which results in the most reliable information. It is often necessary, however, to design and build the flight control system before any such operational testing can be performed, since the vehicle may be unflyable without the control system. Prohibitive costs may also make such testing impractical. Thus, the data on the vehicle dynamics which is available to the control system designer may contain considerable inaccuracies.

The characteristics of the bending motion of flight vehicle is also of great importance in the control system design, since special means of compensation is often required to insure the stability of the bending modes. Typically, the mode shapes and natural frequencies of the bending motion are not known accurately, since it is difficult and often impractical to determine these experimentally. These parameters are also likely to be subject to changes when varying flight conditions are encountered. Variations in the performance of control system components due to normal tolerances is another source of system uncertainty. The effects of these variations can be controlled to some extent by specifying their permissible range, but it is often required that standard off-the-shelf components be used, in which case the tolerances are imposed on the design. Common variations of this type are changes in static sensitivity, which are usually accounted for by specifying a minimum gain margin in order to prevent instability. Similarly, phase margins have been used to insure stability despite changes in the dynamic characteristics of the system components.

5.4 Booster Attitude Control System

The characteristics of the vehicle, which will be considered here, were obtained from Reference [36]. This vehicle is of particular interest because of the low damping and natural frequency of the structural bending motion, which is sensed by the on-board instruments and is fed back through the control system to the engine actuators.

Only the first bending mode is included in the vehicle dynamics, which are described by the following equations of motion:

$$\ddot{\theta} - .0733\,\alpha + .45\,\beta = 0$$

$$\dot{\alpha} - \dot{\theta} + .0405\,\theta + .01067\,\alpha + .02106\,\beta = 0$$

$$\ddot{\eta} + 2(.005)\,\omega_b\dot{\eta} + \omega_b^2\eta = 15.83\,\beta \qquad (5.6)$$

$$\dot{\beta} + 17.9\,\beta = \beta_c$$

$$\beta_c = \beta_c(\theta_c, \theta_i, \dot{\theta}_i)$$

where the symbols represent these quantities:

θ = pitch angle of the rigid vehicle (rad.)

θ_i = pitch angle measured by attitude gyro (rad.)

$\dot{\theta}_i$ = pitch rate measured by rate gyro (rad./sec.)

θ_c = commanded pitch angle (rad.)

α = angle-of-attack (rad.)

β = engine gimbal angle (rad.)

β_c = commanded gimbal angle (rad.)

η = displacement of bending mode (m.)

ω_b = bending mode frequency (rad./sec.)

ξ_b = damping ratio of bending mode

The positive direction of these angles and the bending displacement are defined in Figure 5.1. The rigid body motion is described by the first two equations and control torques are exerted by deflection of the thrust through the gimbal angle, β. The last expression indicates that the commanded gimbal angle depends on the commanded pitch angle, which is supplied by the guidance system, and the measured pitch angle and pitch rate. The form of the relationship remains to be determined. The bending mode displacement is assumed to be excited only by the deflection of the thrust vector, neglecting the aerodynamic

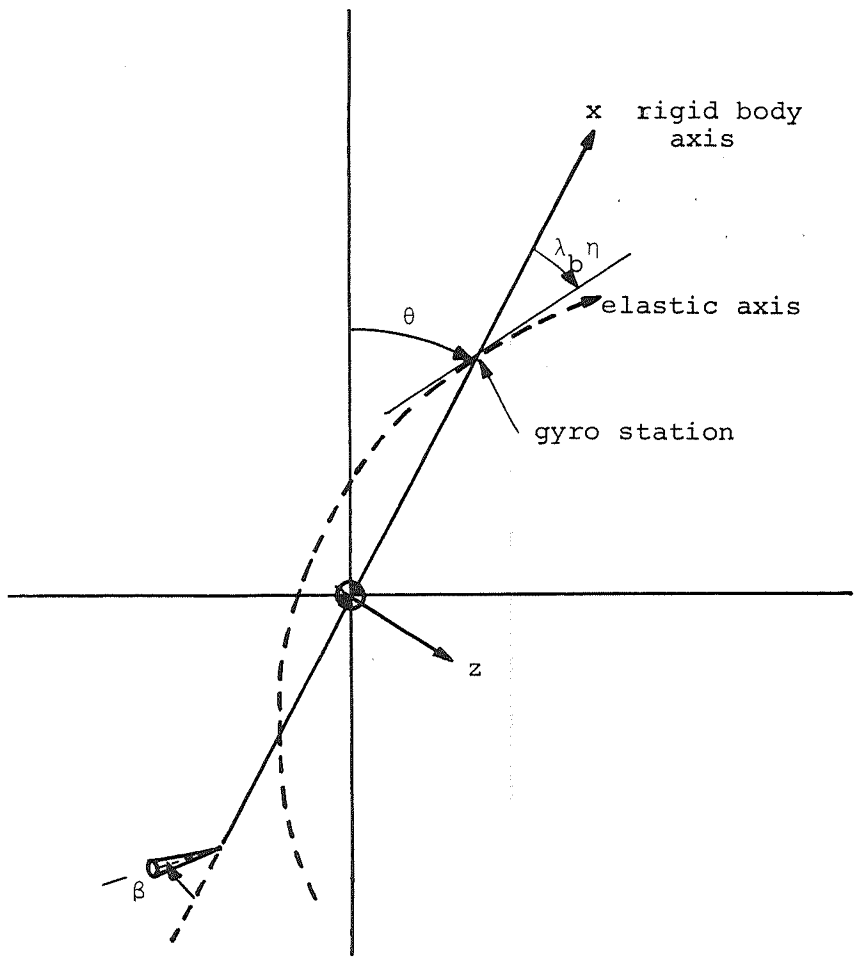


Figure 5.1. Geometry of elastic booster

forces on the body. The actuator dynamics, relating the commanded and actual gimbal angles are approximated by a first order lag. Both pitch angle and pitch rate are measured by gyros located at the same station in the vehicle. Their outputs are given by the following expressions:

$$\theta_i = \theta + \lambda_b \eta$$

and

$$\dot{\theta}_i = \dot{\theta} + \lambda_b \dot{\eta}$$

where

λ_b = slope of the bending mode at the instrument station (rad./m.)

The values of the bending frequency and mode slope are assumed to be inaccurately known with a normal distribution about their mean values. They are, furthermore, taken to be uncorrelated with the following statistics:

$$\overline{\omega_b} = 2.317 \text{ rad./sec.}$$

$$\overline{\lambda_b} = 0.02 \text{ rad./m.}$$

$$\overline{(\omega_b - \overline{\omega}_b)^2} = (0.05)^2 \, \overline{\omega}_b^{\,2}$$

$$\overline{(\lambda_b - \overline{\lambda}_b)^2} = (0.1)^2 \, \overline{\lambda}_b^{\,2}$$

The standard deviation of these parameters is, therefore, 5% and 10%, respectively.

The nominal response of the system to commanded changes in the pitch angle should be as fast as can be practically achieved with an overshoot of no more than 20%. The response should furthermore remain stable for at least two standard deviations of each of the structural parameters while the other is held constant at its mean value.

The transfer function relating the rigid body pitch angle to the gimbal angle can be determined from the equations of motion as:

$$\frac{\theta(s)}{\beta(s)} = \frac{.0733\ (s+.014)}{(s-.0411)(s+.294)(s-.242)} \tag{5.7}$$

Since only the short period dynamics are of interest here the pole and zero close to the origin may be cancelled assuming that the stability of the cancelled mode will be considered separately. This gives:

$$\frac{\theta(s)}{\beta(s)} = \frac{.0733}{(s+.294)(s-.242)} \tag{5.8}$$

The transfer function of the bending mode displacement is obtained from Equation (5.6) as:

$$\frac{\eta(s)}{\beta(s)} = \frac{15.83}{s^2 + 2(.005)\omega_b s + \omega_b^2} \tag{5.9}$$

The time constant of the gimbal actuator is very small and has a negligible effect on the system response. The gimbal angle is, therefore, assumed to be equal to the commanded gimbal angle:

$$\beta(s) \cong \beta_c(s)$$

The transfer functions of the rigid vehicle and the bending motion can now be added in order to give a single pitch angle transfer function for the flexible vehicle. With minor approximations this transfer can be expressed as:

$$\frac{\theta_i(s)}{\beta(s)} = \frac{-15.83\ \lambda_b(s^2+.052s-.0046\ \frac{\omega_b^2}{\lambda_b})}{(s+.294)(s-.242)(s^2+2(.005)\omega_b s+\omega_b^2)} \tag{5.10}$$

This transfer function gives the relationship between the pitch angle sensed by the attitude gyro and the gimbal deflection. The block diagram of the attitude control system is shown in Figure 5.2, where pitch rate is fed back, in addition to the attitude feedback, in order to stabilize the rigid body mode. The root-locus of this system without any compensation in the forward path is plotted in Figure 5.3 for the nominal values of ω_b and λ_b with the gain of the pitch rate feedback equal to unity.

If no compensation is included in the system it is clear that the bending mode is unstable for all practical loop gains. A second order filter can be used to improve the bending mode behaviour by giving the proper amount of phase-shift at the bending frequency. The transfer function of this filter is:

$$G_f(s) = \frac{\omega_f^2}{s^2 + 2(.707)\omega_f s + \omega_f^2} \tag{5.11}$$

where the damping ratio has been chosen, but its natural frequency, ω_f, will be optimized by the design procedure. The root-locus of the system with the bending filter included is also shown in Figure 5.3 for a single value of ω_f. It is seen that considerable improvement can be achieved in the damping ratio of the bending mode by using this compensation.

variable design parameters:

ω_b - bending frequency ; s.d.=0.05 $\overline{\omega_b}$

λ_b - bending mode slope ;s.d.=0.10 $\overline{\lambda_b}$

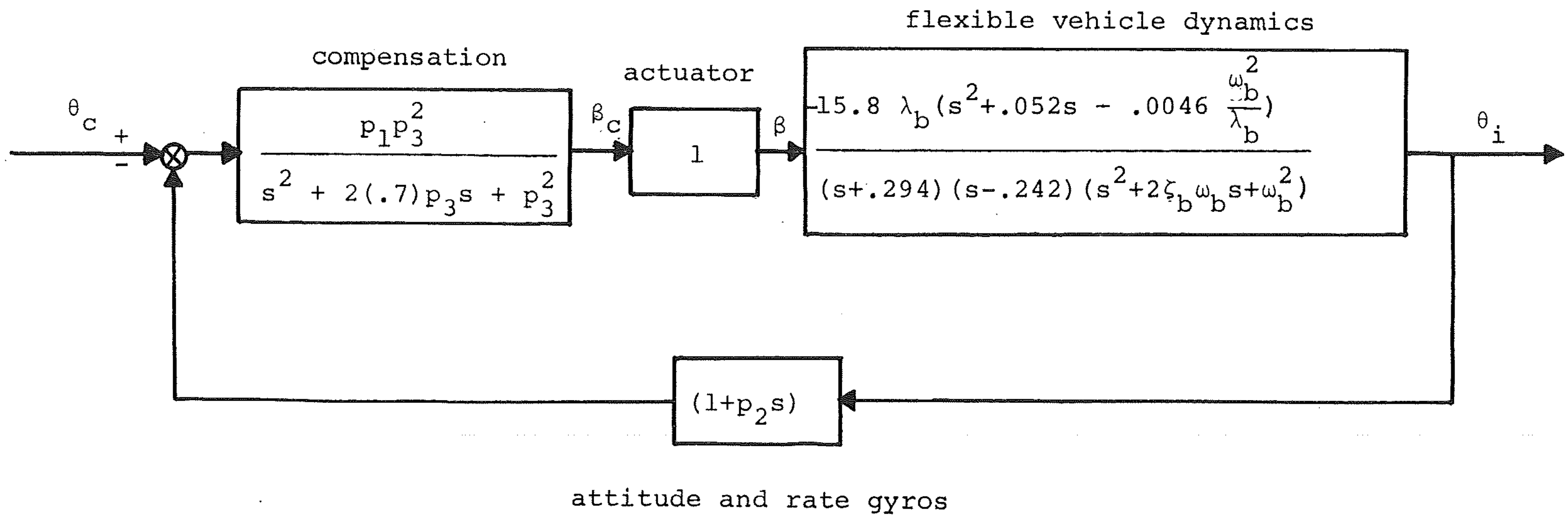


Figure 5.2 Block diagram of booster attitude control system

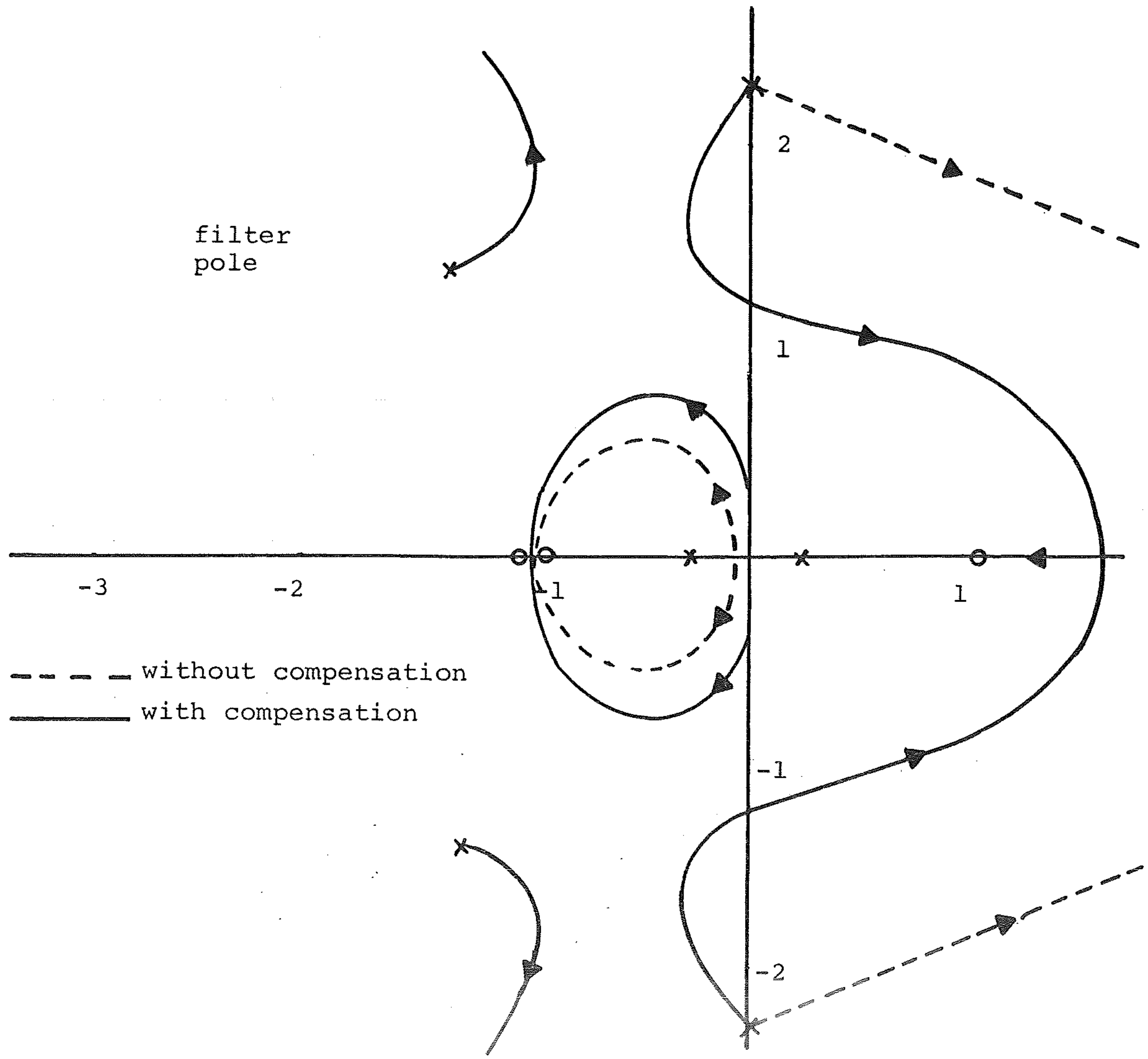


Figure 5.3. Root locus of attitude control system.

In addition to the natural frequency of the bending filter two other free design parameters are specified. These are the static sensitivity of the compensation and the amount of pitch rate feedback, denoted by p_1 and p_2 respectively.

In order to apply the model performance index to this problem it is necessary to select the transfer function of a reference model, whose characteristic coefficients are then used to compute the weighting matrix of the system states. This model represents the desired response characteristics of the total system to a commanded change in the pitch angle. The transfer function chosen for this purpose is of fourth order and is given by:

$$\frac{\hat{\theta}_i(s)}{\theta_c(s)} = \frac{(1.8)^2}{(s^2+2(.707)s+(1)^2)(s^2+2(.2)1.8s+(1.8)^2)} \quad (5.12)$$

where $\hat{\theta}_i$ is the pitch angle response of the model as it would be measured at the gyro station. The first of the two second order modes of the model corresponds to the desired rigid vehicle mode and has been chosen to be well damped with a natural frequency of 1 rad./sec. The root-locus of Figure 5.3 indicates that the closed-loop poles of the vehicle can achieve this damping ratio and natural frequency. The system transfer function is non-minimum phase, however, and cannot be expected to respond as fast as the model, since the zero in the right half complex plane has a time delaying effect on the response.

The second mode of the model transfer function represents the desired bending mode characteristics. A rather moderate value has been chosen for the damping ratio in order to prevent undue emphasis on stabilizing the bending mode. The natural frequency was chosen somewhat lower than the natural bending frequency, recognizing the fact that the corresponding closed-loop system mode has a tendency towards lower frequencies, as can be seen from the root-locus of Figure 5.3.

From Figure 5.2 it can be determined that the system closed-loop transfer function has six poles and two zeros. Since the model is fourth order and contains no zeros, both system and model have the same number of excess poles over zeros, which results in a constant weighting matrix according to the development of Section 3.5. The form of the weighting matrix is given by Equation (3.86):

$$\underline{Q} = \underline{\tilde{\alpha}}^T \, \underline{\tilde{\alpha}}$$

where $\underline{\tilde{\alpha}}$ is an n-dimensional vector containing the coefficients of the model's characteristic equation, which in this case becomes:

$$\underline{\tilde{\alpha}}^T = [\ 3.24,\ 5.305,\ 5.258,\ 2.134,\ 1.0\ ]$$

The performance index, which is to be minimized, is then written as:

$$\overline{J} = \int_0^\infty \underline{x}_*^T \, \underline{Q} \, \underline{x}_* \, dt + \varepsilon^2 \int_0^\infty \overline{\delta \underline{x}^T \, \underline{Q} \, \delta \underline{x}} \, dt \tag{5.13}$$

where the system state and its deviation are described by Equations (3.5) and (3.15), respectively. Only the roots and static sensitivities of the open-loop transfer functions have to be provided to the computer programs of Appendix C, which then compute the required closed-loop coefficients.

It is convenient to use a weighting constant, ε^2, to change the emphasis on the sensitivity index relative to the nominal part of the performance index. Thus, when $\varepsilon=1$, $\bar{J}$ represents the expected value of the model performance index for the specified parameter covariance matrix, which determines the magnitude of the second integral of $\bar{J}$. Setting $\varepsilon=2$, for instance, is completely equivalent to multiplying the parameter covariance matrix by the square of ε, as can be shown by using the linear relationship between $\delta \underline{x}$ and the parameter variations, $\delta \underline{\xi}$. The effect of the parameter uncertainties on the performance index can, therefore, be changed through the value of ε without disturbing the interrelationship between their variations.
The performance index may then be written as:

$$\bar{J} = J_* + \varepsilon^2 J_s$$

and the trade-off parameter, μ, becomes:

$$\mu = 1 - \frac{1}{\varepsilon^2} \frac{|\Delta J_*|}{|\Delta J_s|}$$

5.4.1 Sensitivity to Bending Frequency Variations

The design method is first applied to the problem considering only the effects of uncertainties in the bending mode frequency. Thus:

$$\xi = \omega_b$$

$$\xi_* = \overline{\omega_b} = 2.317 \text{ rad./sec.}$$

$$R = \overline{\delta\xi^2} = 0.0025 \cdot \overline{\omega_b}^2$$

The computer programs in Appendix C were used to minimize the performance index of Equation (5.13), starting with the weighting factor $\varepsilon=0$, which results in the model performance index design based on the nominal value of ω_b. Next the weighting factor was increased somewhat arbitrarily to $\varepsilon=6$, which leads to the minimum expected value of the model performance index for six times the specified variance of ω_b. The corresponding values of the free design parameters are given in Table 5.1.

It can be seen from this table that the effect of including the sensitivity index in the performance index is to decrease the static sensitivity, increase the rate feedback, and decrease the natural frequency of the bending filter, when compared with the solution based on the nominal value. These changes are all relatively small, but it is interesting to compare the values of the nominal performance index and the sensitivity index for these design solutions.

	design number		
parameter	1	2	3
ε	0	6	12
p_1	2.48	2.16	1.99
p_2	2.12	2.32	2.40
p_3	1.58	1.40	1.36
J_*	2.05	2.31	2.54
J_s	0.062	0.0071	0.0039
$\bar{J}$	2.05	2.56	3.10
μ		0.87	0.77

Table 5.1 Values of free design parameters and performance indices, with uncertainties in ω_b

These values are listed in Table 5.1, which also gives the values of μ for the two designs with non-zero weighting of the sensitivity index.

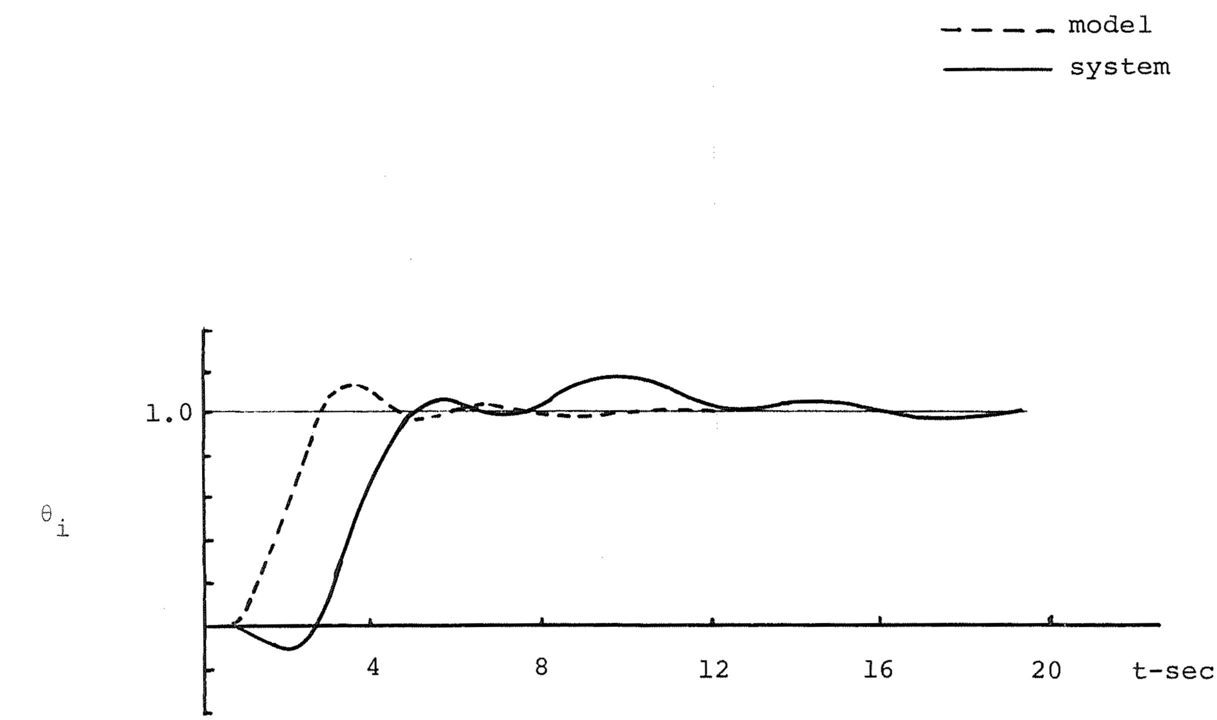
Design no. 1 represents the results of minimizing J_* without any regard for sensitivity. Comparing the values of the performance indices for this design with those of design no. 2, it is clear that the minimization of $\bar{J}(\varepsilon=6)$ has the effect of reducing the value of J_s by an order of magnitude, at the same time as the value of J_*, representing the nominal system performance, is increased by a much smaller amount.

It was suggested in Section 5.3 that the ratio denoted by μ and defined in terms of the changes in J_* and J_s, can be used as an index of the systems' ability to reduce its sensitivity to a specified parameter variation, using the sensitivity of the nominal design as a reference. For design no. 2, this ratio is 0.87, since $0 \leq \mu \leq 1$ with the lower limit indicating no possible improvement in the sensitivity index (or $\bar{J}$), this system may be rated as responsive to reduction in sensitivity to the parameter under consideration.

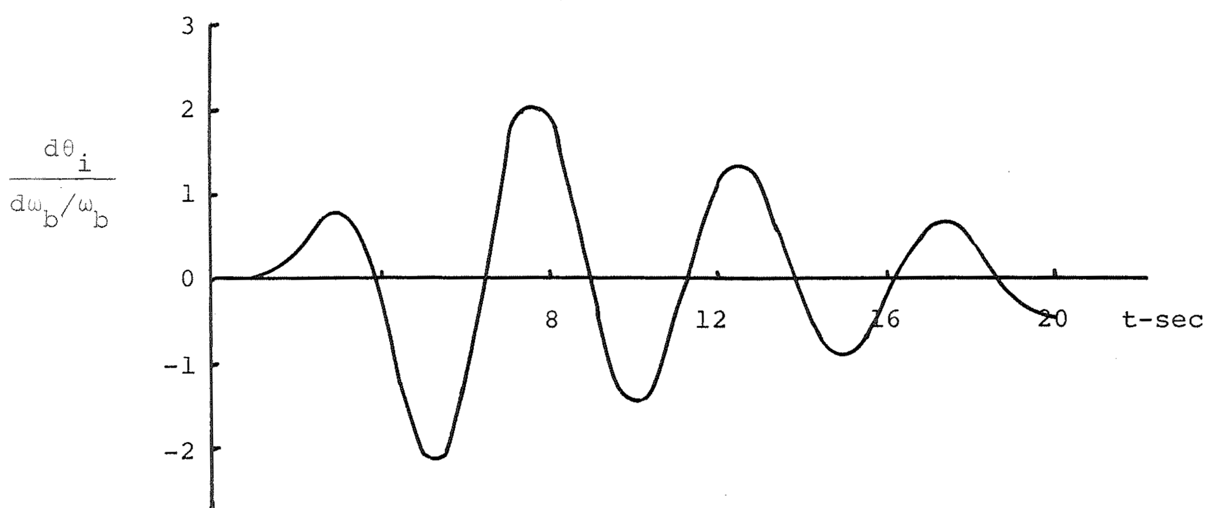
The sensitivity of the system as measured by J_s can be reduced even further by increasing the weighting coefficient of the sensitivity term. The results for $\varepsilon=12$ are given in Table 5.1 as design no. 3. The effect on the free design parameters is the same as before with further decrease of the static sensitivity, a slight increase in rate feedback and reduction of the filter frequency. The reduction in J_s from 0.0071 to 0.0039 in going from design no. 2 to design no. 3 is considerable, although nowhere as significant as obtained by design no. 2. The price that must be paid in terms of a deteriorating nominal performance has also become higher per unit improvement in J_s as indicated by the change in J_* when compared for designs no. 2 and no. 3. This fact is also reflected in the value of μ for design no. 3, which has decreased somewhat due to the effect of diminishing returns.

These relative changes in the performance indices must, however, be interpreted in terms of the time responses of the corresponding system designs in order to be meaningful. The normalized response of design no. 1 to a step input in commanded pitch angle is shown in Figure 5.4. The pitch angle response of the system is similar to the model's response except for the time delaying effect of the non-minimum phase characteristics. The overshoot is 17% which is within the 20% limit and the settling time to within 5% of the steady-state output is 11.8 sec. The same time delaying effect is noted in the pitch rate response, which is similar to the model's response in other respects. The sensitivity functions corresponding to pitch angle and pitch rate is also shown in Figure 5.4. These indicate a strong tendency towards an oscillatory response with changes in the bending frequency, ω_b. The root-locus in Figure 5.5 shows that the structural mode has been well damped but the mode corresponding to the bending filter has a damping ratio of only 0.15. The period of the sensitivity functions indicate that this mode may be adversely affected by changes in the structural frequency.

The effect of decreasing ω_b by 10% from its nominal value is given by Figure 5.6 which shows that the system response is unstable

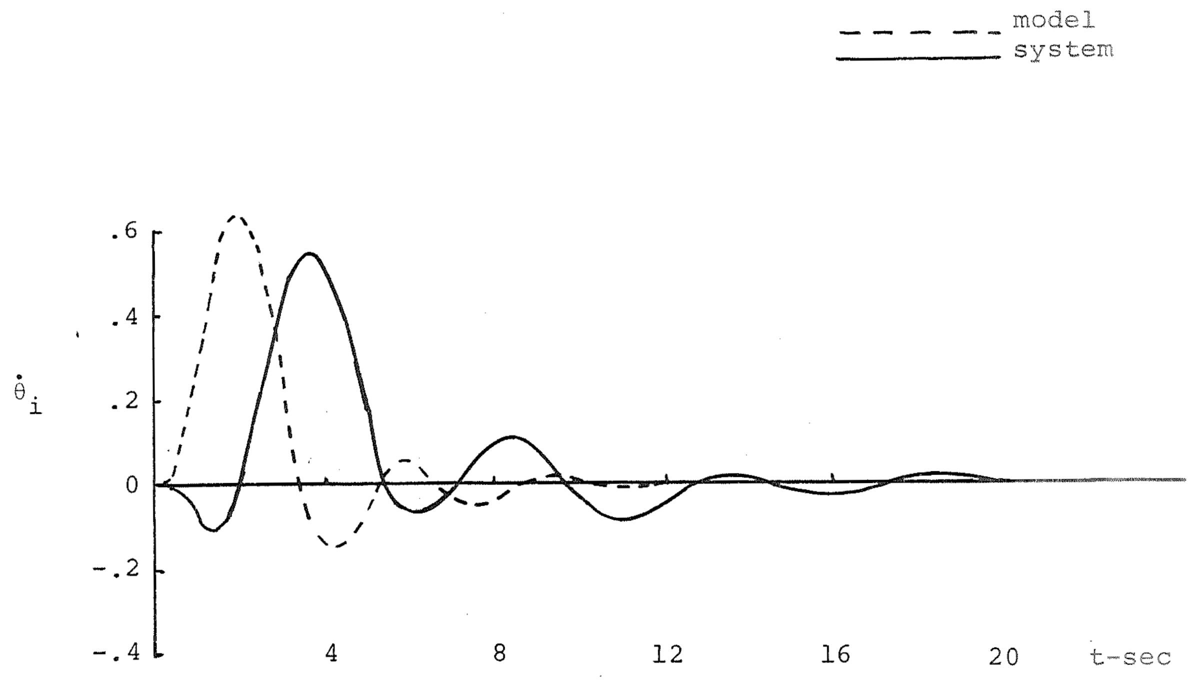


Pitch response at gyro station , $\omega_b = \omega_b^*$

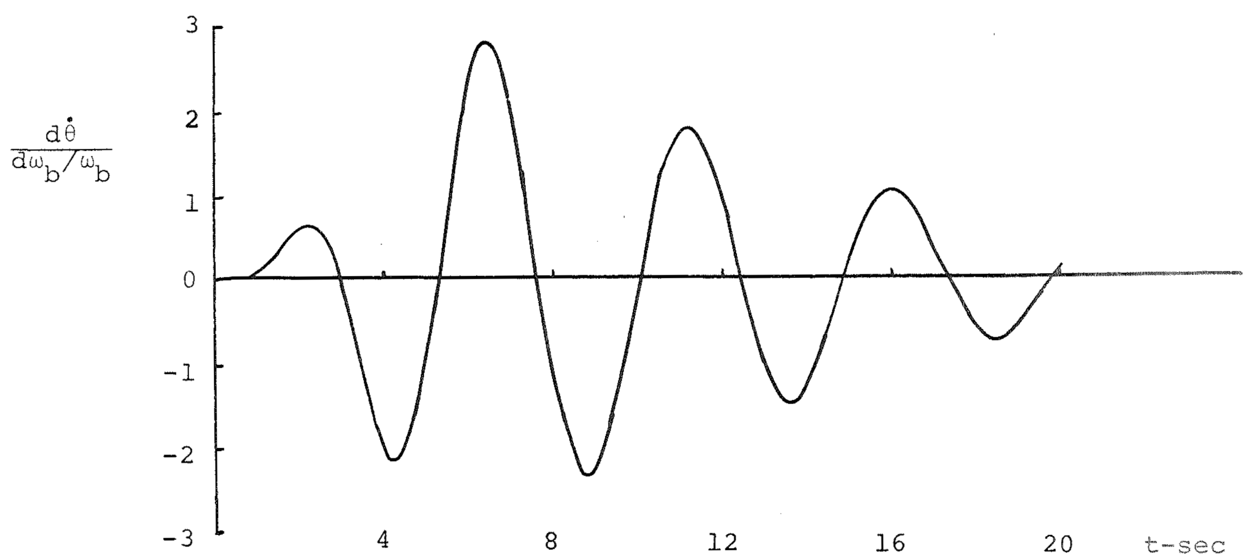


Sensitivity of pitch response w.r.t. ω_b

Figure 5.4. Step response of design no. 1 (continued)



Pitch rate response at gyro station



Sensitivity of pitch rate response

Figure 5.4. Step response, design no. 1

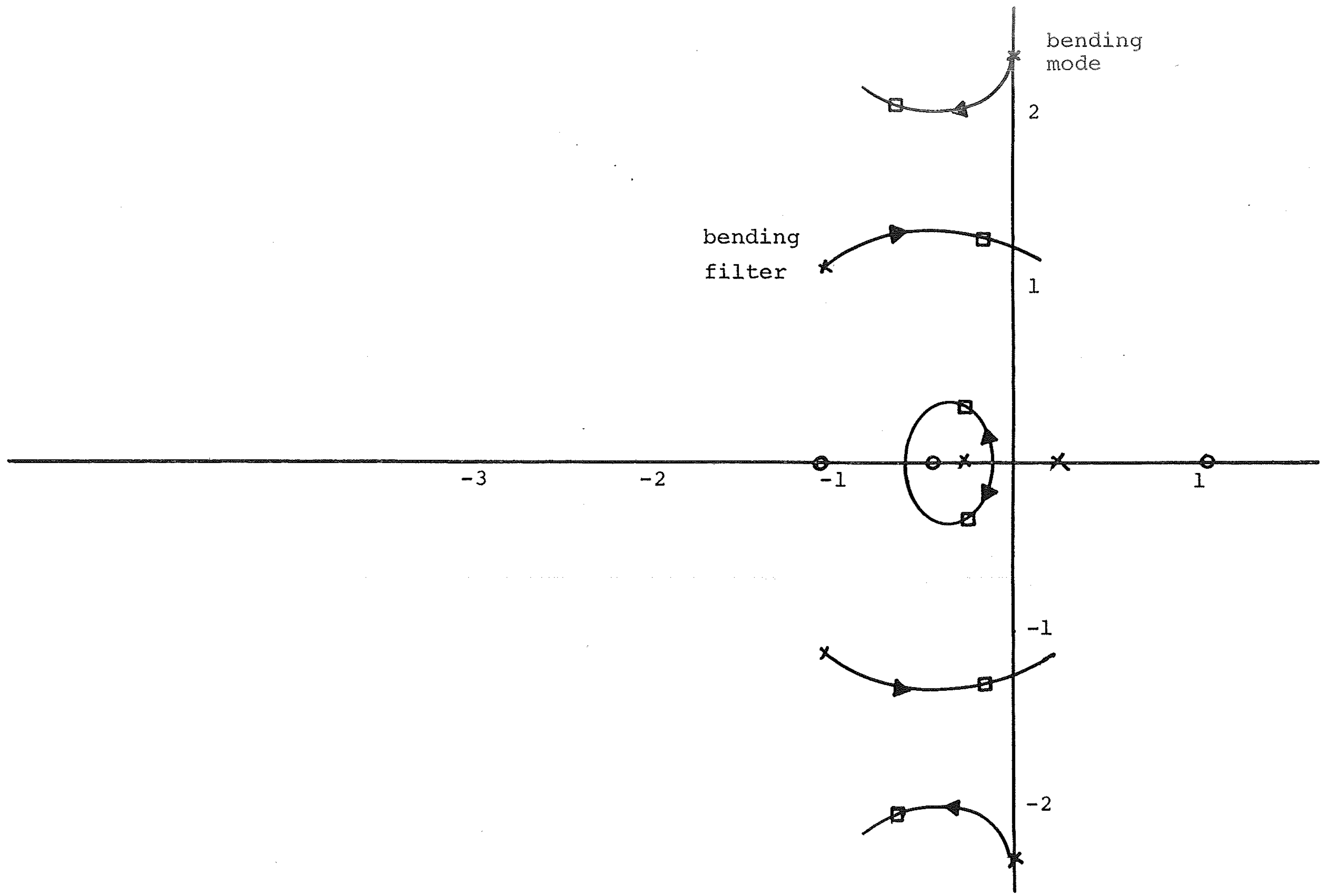


Figure 5.5. Root locus of design no. 1

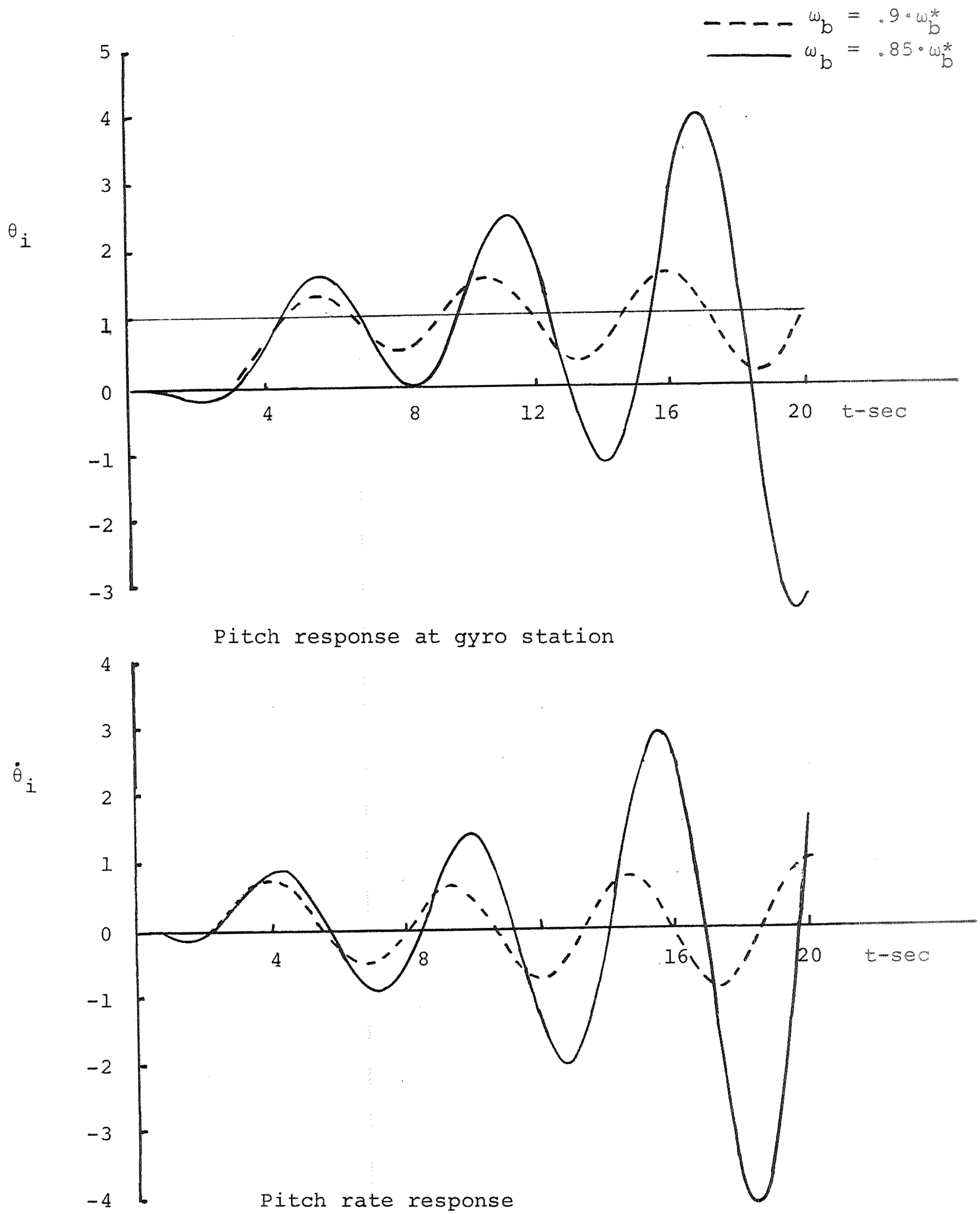
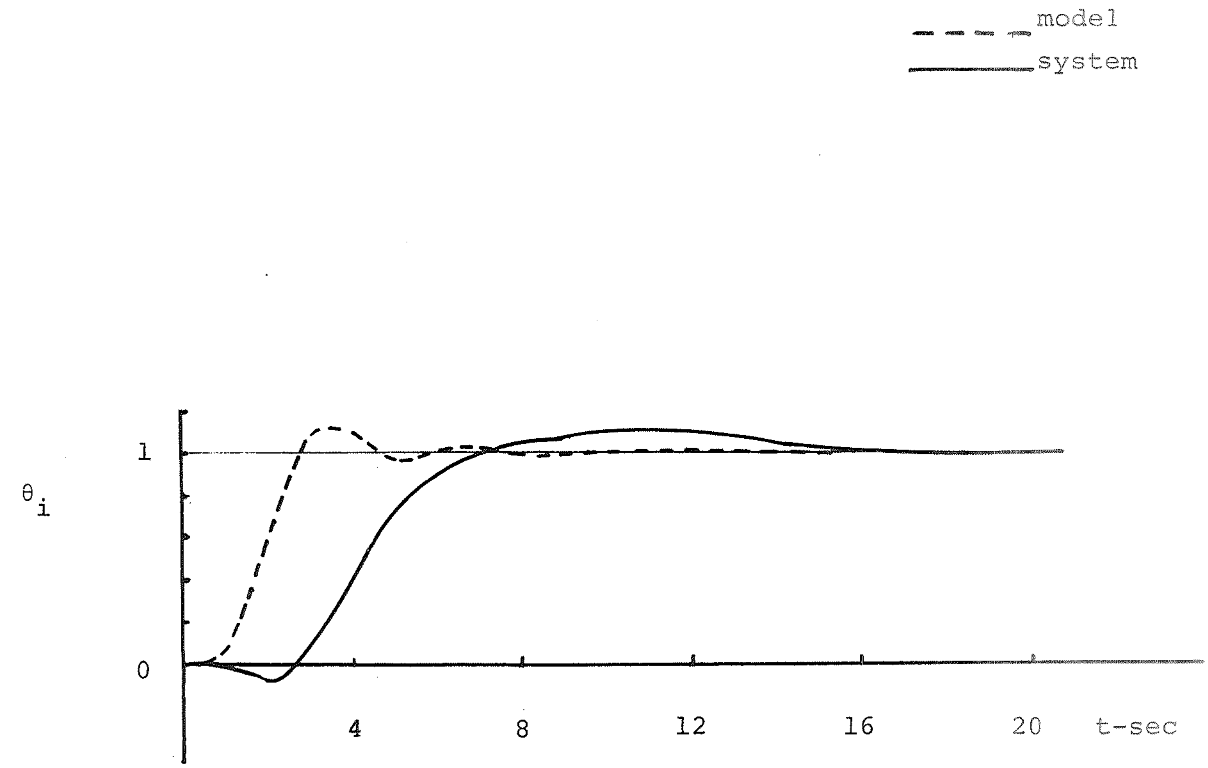


Figure 5.6. Off-nominal response, design no. 1.

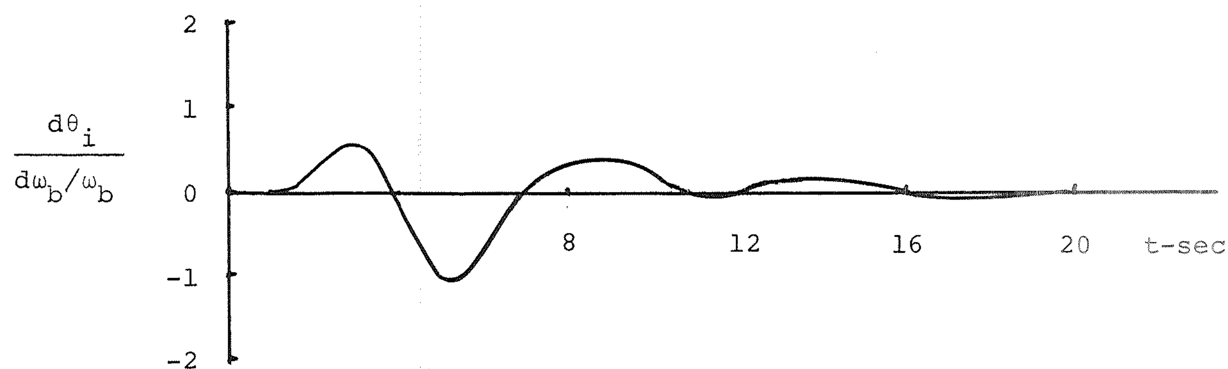
for this value of the bending frequency. A 15% reduction in ω_b is seen to result in a fast divergence of the response. Design no. 1 is, therefore, unacceptable, since it does not satisfy the requirements on stability when ω_b changes by two standard deviations. The case of increasing ω_b causes no difficulties as could be expected from the fact that the poles of the bending mode move further away from the bending filter poles.

The response for design no. 2 is shown in Figure 5.7. The pitch response has slightly less overshoot than design no. 1, but its settling time is somewhat longer or about 13.0 sec. This is reflected in the pitch rate whose peak is reduced when compared with design no. 1. The really significant effect is noted in the sensitivity responses which have much smaller amplitudes than before and increased damping. The root-locus plot in Figure 5.8 also indicates that the damping ratio of the bending filter mode has been increased to 0.27 with a slight decrease in the damping of the bending mode. The importance of these differences in designs no. 1 and no. 2 are shown by the off-nominal responses in Figure 5.9. For a 10% decrease in ω_b, the response of design no. 2 is clearly stable and, furthermore, the pitch angle response still satisfies the specifications of less than 20% overshoot. A 15% reduction of ω_b puts this design on the verge of instability, but the residue of the unstable mode is significantly less in this case than for design no. 1, which means that the approach of instability will be much less severe for design no. 2.

Thus, design no. 2 satisfies the requirement of a stable response for a 10% deviation of ω_b with a comfortable margin and a relatively smooth response. The pitch response of design no. 3 is shown in Figure 5.10. The increased emphasis on the sensitivity of this design is seen to further reduce the amplitude of the sensitivity response which is obtained at the expense of an increase in the



Pitch rate response at gyro station



Sensitivity of pitch rate response w.r.t. ω_b

Figure 5.7. Step response, design no. 2.(continued)

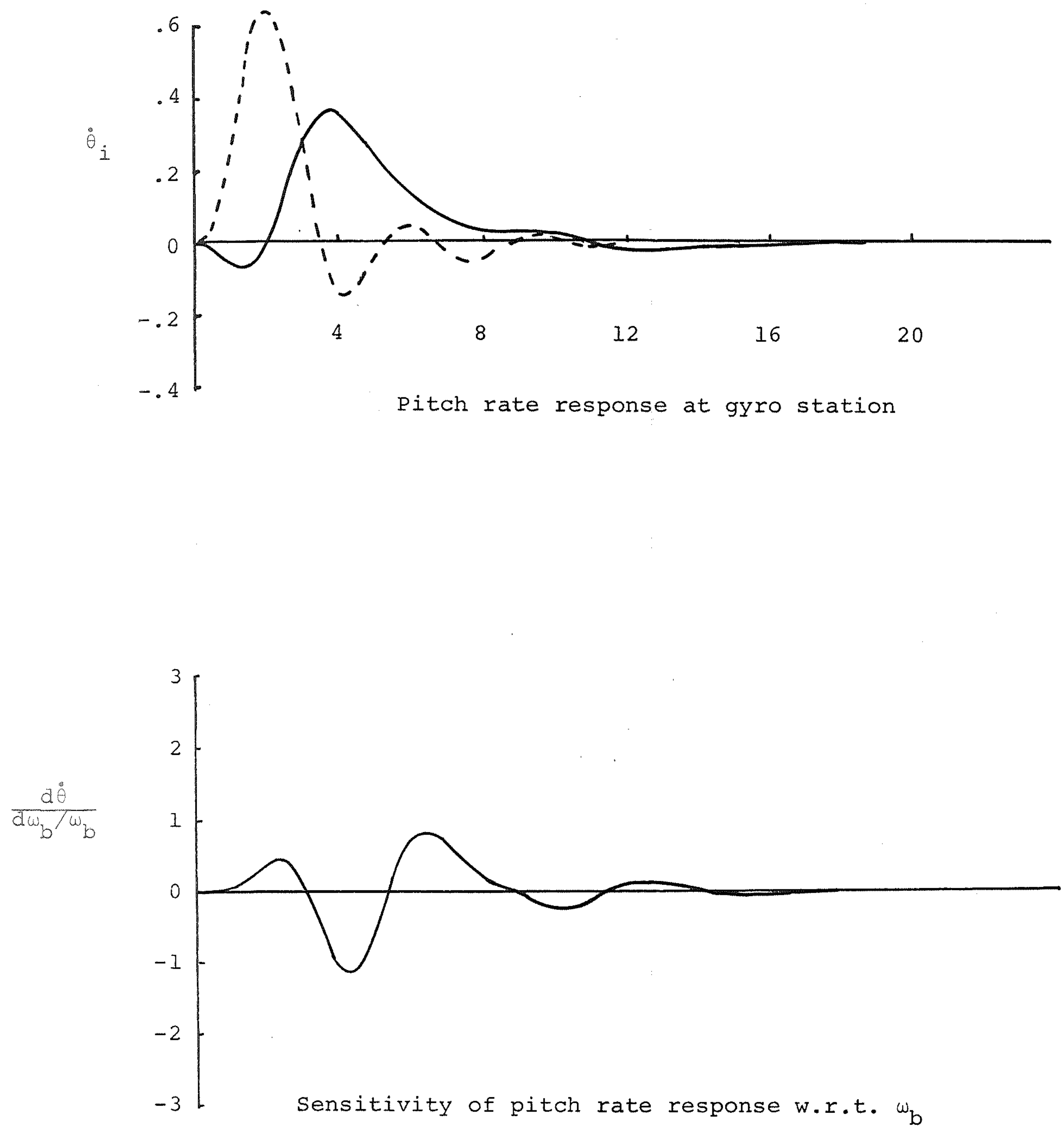


Figure 5.7. Step response, design no. 2

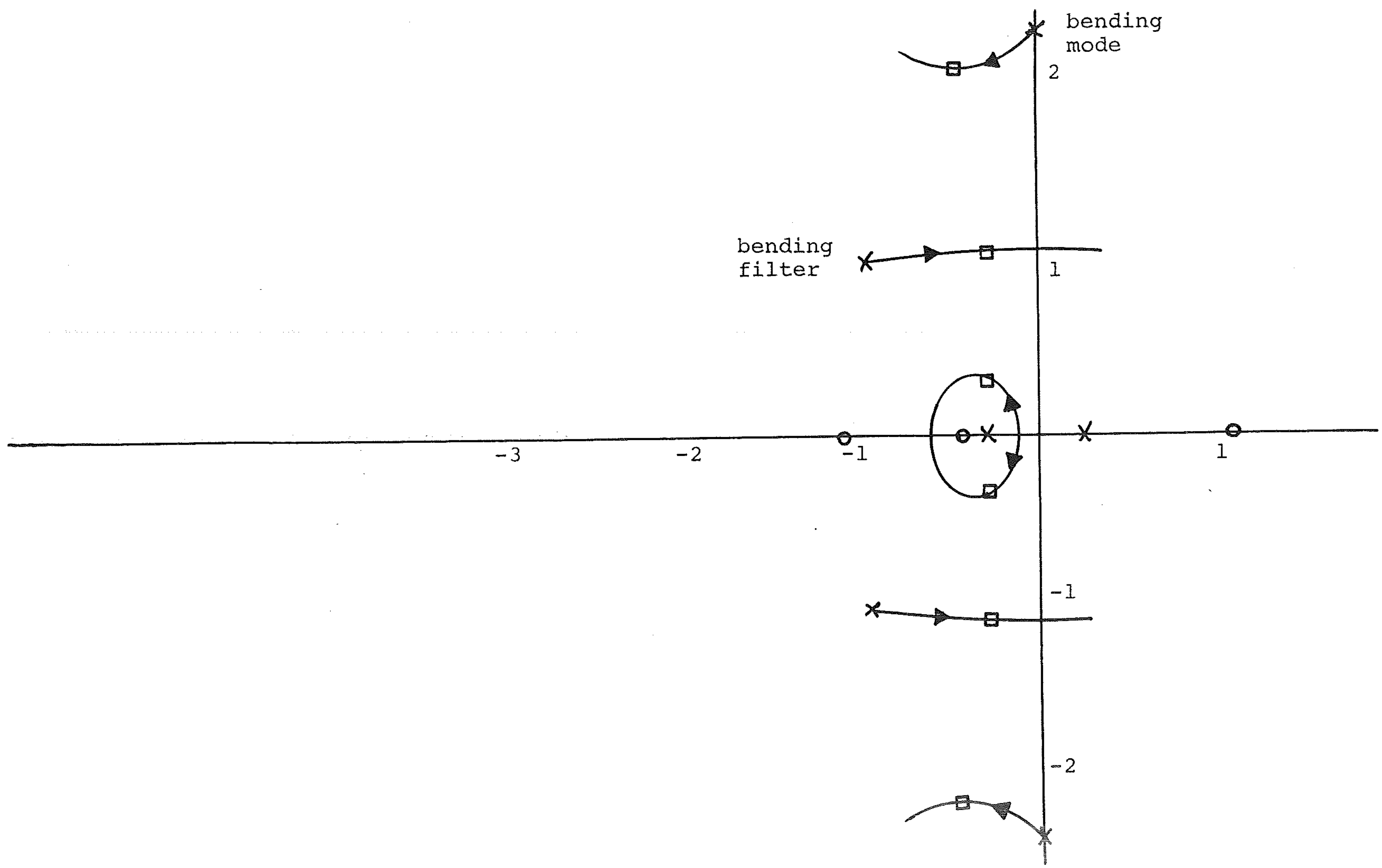
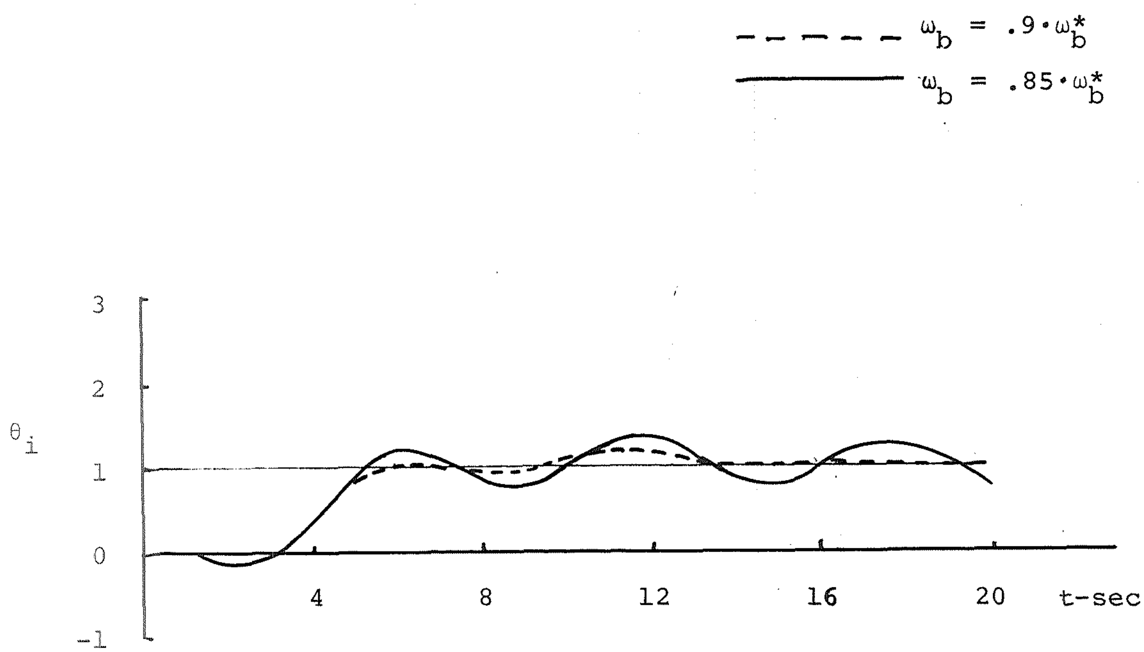
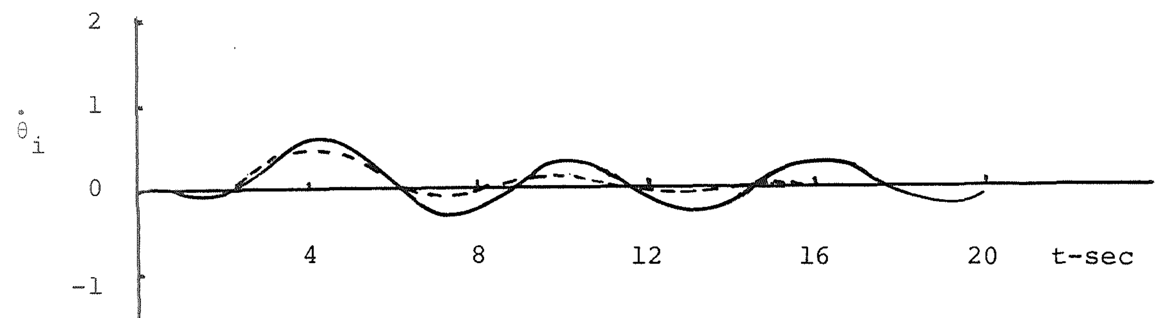


Figure 5.8. Root locus of design no. 2

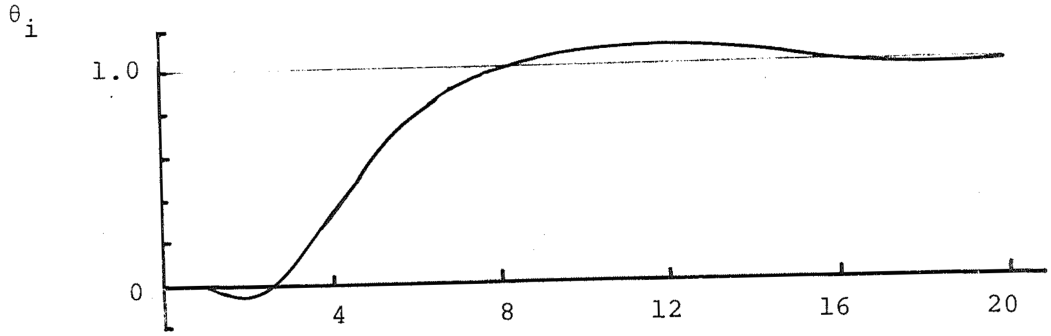


Pitch response at gyro station

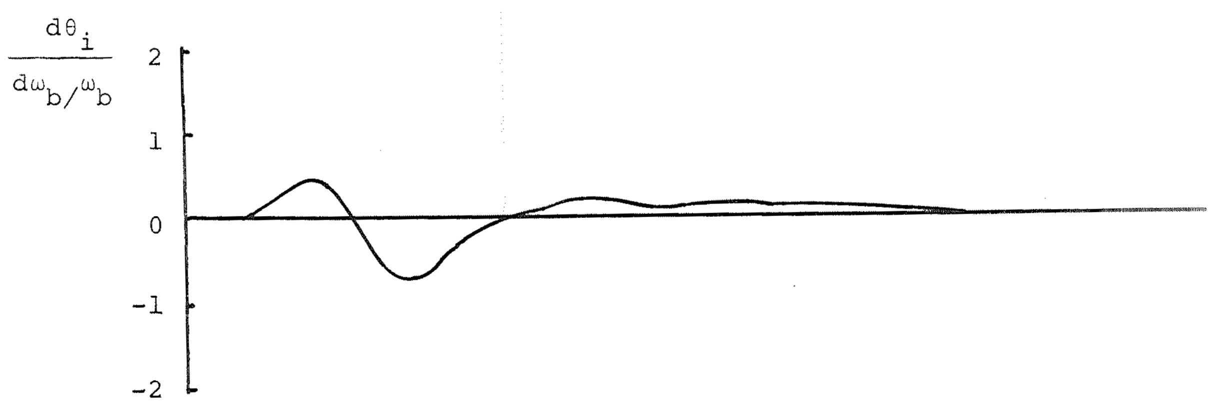


Pitch rate response at gyro station

Figure 5.9. Off-nominal response of design no. 2

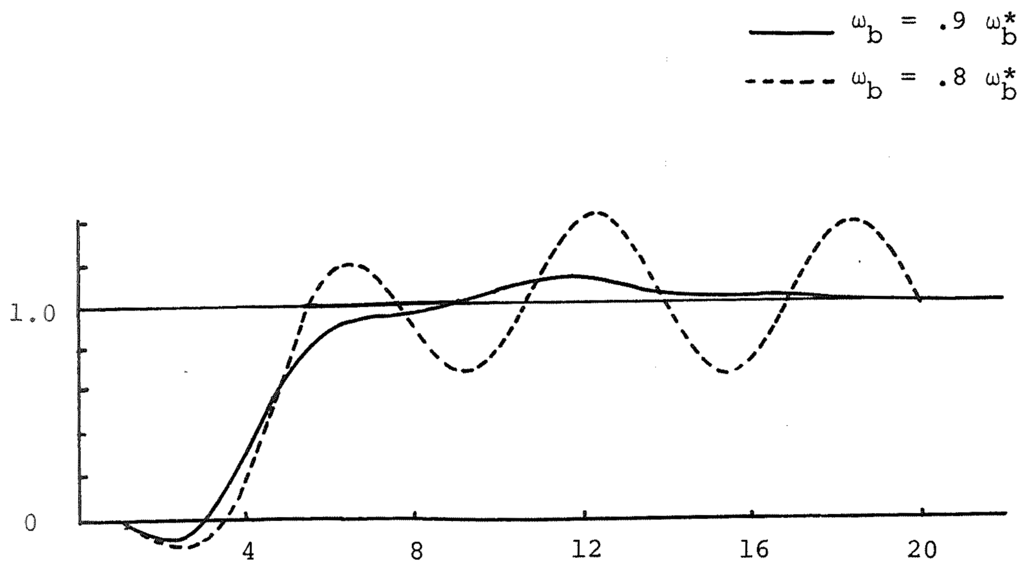


Pitch response at gyro station



Sensitivity of pitch response w.r.t. ω_b

Figure 5.10. Step response of design no. 3.



Pitch response at gyro station

Figure 5.11. Off-nominal responses of design no. 3

settling time to 15.6 sec. as compared with 13.0 sec. for design no. 2. The off-nominal responses of this design are shown in Figure 5.11 for decreasing bending frequency. For a 10% decrease, the response is somewhat better damped than the corresponding response for design no. 2. A 20% variation in ω_b is now required in order to drive the system unstable. Design no. 3, therefore, meets the requirements on stability with a wider margin than design no. 2.

5.4.2 Sensitivity to ω_b and λ_b

Addition of the slope of the bending mode as an uncertain parameter to the design process is easily accomplished using the present method. The vector of variable parameters becomes:

$$\underline{\xi}^T = [\ \omega_b,\ \lambda_b\ ]$$

with a nominal value:

$$\underline{\xi}_*^T = [\ 2.317,\ 0.02\ ]$$

and the covariance matrix:

$$\underline{R} = \begin{bmatrix} .0025\ \overline{\omega_b^2} & 0 \\ 0 & .01\ \overline{\lambda_b^2} \end{bmatrix}$$

since the uncertainties of ω_b and λ_b are uncorrelated. Using a weighting factor of $\varepsilon=6$ as before, the performance index was minimized for this value of the covariance matrix. The solution is referred to as design no. 4 whose parameter and index values are listed in Table 5.2.

	design number		
parameter	1	2	4
ε	0	6	6
p_1	2.48	2.16	2.15
p_2	2.12	2.32	2.27
p_3	1.58	1.40	1.40
J_*	2.05	2.31	2.34
J_s	0.075	0.0098	0.0088
$\bar{J}$	2.05	2.664	2.660
μ			0.86

Table 5.2 Values of free design parameters and performance indices with uncertainties in both ω_b and λ_b

By comparison of the free design parameters for design no. 4 with those of design no. 2, it is seen that these two designs can be assumed to be identical for all practical purposes. The effect of the uncertainty in λ_b on the sensitivity index may be determined for designs no. 1 and 2 as the difference between the values of J_s in Tables 5.1 and 5.2 for each design. This is so because of the independence of the two sources of uncertainty which means that J_s can be written:

$$J_s = J_s\Big|_{\omega_b=\omega_b^*} + J_s\Big|_{\lambda_b=\lambda_b^*} \tag{5.14}$$

where the contribution of each variable is obtained by setting the other variable equal to its nominal value. Thus, for design no. 1 the contribution of the uncertainties in λ_b to J_s is given by:

$$J_s\Big|_{\omega_b=\omega_b^*} = 0.013$$

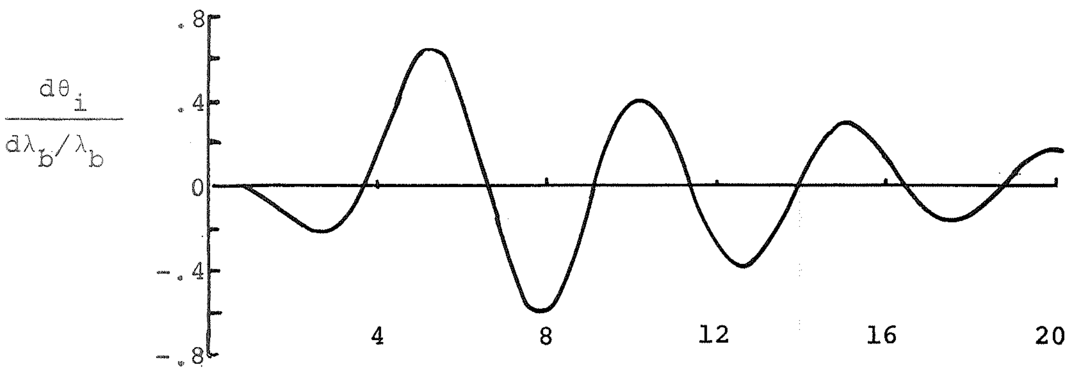
Similarly for design no. 2:

$$J_s\Big|_{\omega_b=\omega_b^*} = 0.0026$$

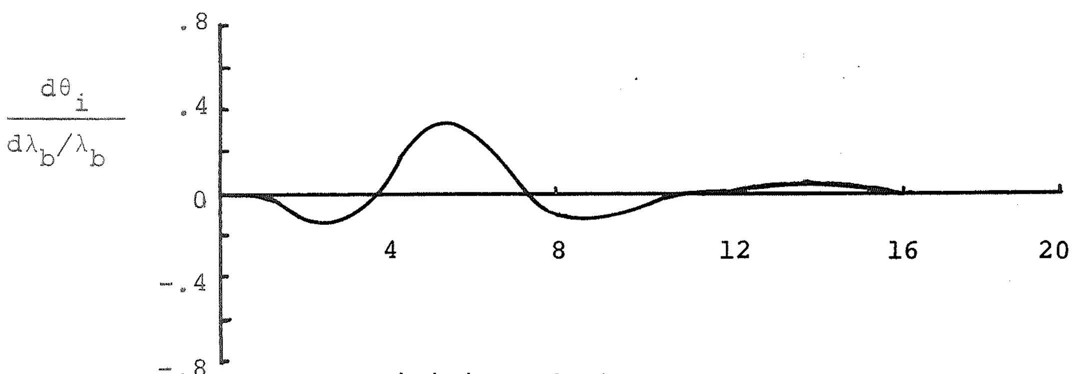
Thus, it is clear that the system is less sensitive to changes in the slope of the bending mode than to changes in the bending frequency, as indicated by the contributions of these two parameters to the sensitivity index. The improvement in the sensitivity of the system to uncertainties in λ_b is, therefore, much less spectacular than the reduction in sensitivity to uncertainties in the bending frequency.

The sensitivity functions of the pitch angle responses of these two designs with respect to λ_b are given in Figure 5.12. The amplitude of this sensitivity function for design no. 1 is much smaller than the amplitude of the sensitivity function with respect to ω_b as may be seen by comparison with Figure 5.4. A given percentage variation of ω_b may be estimated to result in almost four times as large a deviation of the output as the same percentage variation of λ_b. This difference in sensitivity to ω_b and λ_b is considerably less for design no. 2.

Comparison of the two sensitivity functions in Figure 5.12 indicates a significant reduction in sensitivity to λ_b in going from design no. 1 to design no. 2. This is also verified by the off-nominal responses for these designs, which are shown in Figure 5.13 for a 20% increase in λ_b from its nominal value. Design no. 1 exhibits a very lightly damped mode, which may be identified as corresponding to the bending filter. The response, furthermore, has a 30% overshoot. Design no. 2 on the other hand has a relatively well damped response with an overshoot of only 11%. This design, therefore, is seen to meet all the specifications on the system despite the specified variations of the bending frequency and bending mode slope.

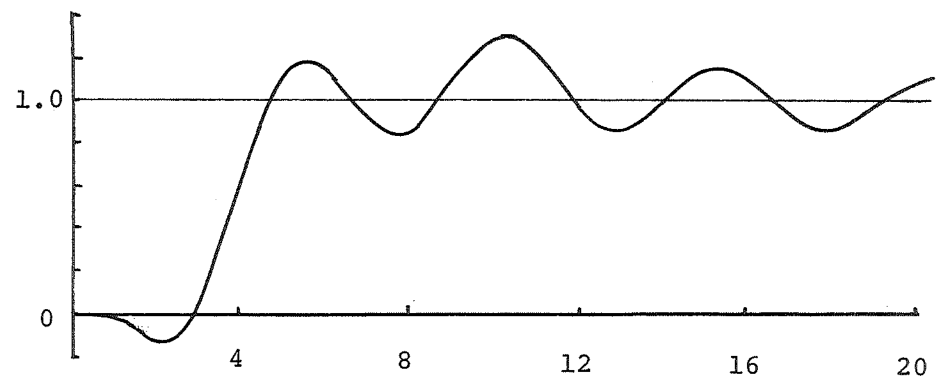


Sensitivity of pitch response, design no. 1

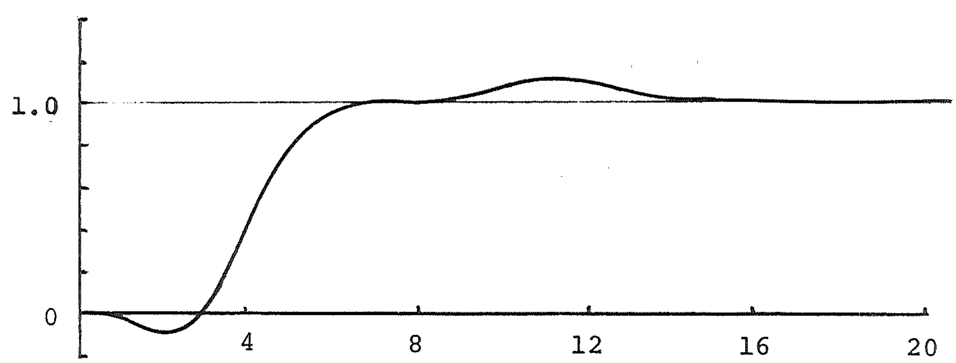


Sensitivity of pitch response, design no. 2

Figure 5.12. Sensitivity functions w.r.t. λ_b



Pitch response, design no. 1



Pitch response, design no. 2.

Figure 5.13. Off-nominal step responses for $\lambda_b = 1.2 \cdot \lambda_b^*$

5.5 Aircraft Attitude Control System

The longitudinal dynamics of a high performance aircraft are described in Reference [44] in a very convenient form for application of the present design method. In particular, the uncertainties in the vehicle's dynamic characteristics are described by the joint distribution of the dimensional coefficients of the equations of motion at each flight condition. The vehicle possesses two lightly damped bending modes which must be stabilized but whose frequencies are inaccurately known. The design of an attitude control system for this vehicle will be considered here for a single flight condition.

The longitudinal short period rigid body dymanics of the aircraft are described by the following set of linearized differential equations:

$$\dot{\alpha} = Z_\alpha \alpha + \dot{\theta} + Z_\delta \delta_e$$

$$\ddot{\theta} = M_\alpha \alpha + M_{\dot{\theta}} \dot{\theta} + M_{\delta_e} \delta_e$$

$$\dot{\delta}_e = -20\delta_e + 20\, \delta_{e_c}$$

$$\delta_{e_c} = \delta_{e_c} (\theta_c, \theta_i, \dot{\theta}_i)$$

where

α = angle-of-attack (rad.)

θ = pitch angle (rad.)

θ_i = measured pitch angle (rad.)

δ_e = elevator deflection angle (rad.)

δ_{e_c} = commanded elevator deflection (rad.)

Z_α, Z_{δ_e}, M_α, $M_{\dot\theta}$ and M_{δ_e} are the dimensional coefficients which are known in terms of a joint normal distribution

$$f(\underline{\xi}) = \frac{1}{(2\pi)^{5/2} \, (|\underline{R}|)^{1/2}} \exp\left[- \frac{1}{2}(\underline{\xi}-\underline{\xi})^T \, \underline{R}^{-1} \, (\underline{\xi}-\underline{\xi})\right] \tag{5.16}$$

where the vector $\underline{\xi}$ consists of the five coefficients:

$$\underline{\xi}^T = [\, Z_\alpha, Z_{\delta_e}, M_\alpha, M_{\dot\theta}, M_{\delta_e} \,]$$

$\underline{R}$ is the covariance matrix of $\underline{\xi}$, given by:

$$\underline{R} = \overline{\delta\underline{\xi} \; \delta\underline{\xi}^T}$$

The mean values of the dimensional coefficients are given as functions of Mach number and dynamic pressure. The covariance matrix $\underline{R}$, is given by Table 5.3 in terms of the mean values of the dimensional coefficients.

The structural response dynamics are described by the equations of the first two bending modes:

$$\begin{aligned} \ddot\eta_1 + 2\xi_{b_1} \omega_{b_1} \dot\eta_1 + \omega_{b_1}^2 \eta_1 &= \psi \, \delta_e \\ \ddot\eta_2 + 2\xi_{b_2} \omega_{b_2} \dot\eta_2 + \omega_{b_2}^2 \eta_2 &= \psi \, \delta_e \end{aligned} \tag{5.17}$$

where η_1 and η_2 are the deflections of the first and second bending modes at a specific reference station. The elevator input coefficient is given by:

TABLE 5.3

DIMENSIONAL COEFFICIENT COVARINANCE MATRIX†

Rows of Covariance Matrix	Columns of Covariance Matrix				
	$(0.150)^2\ \bar{Z}_\alpha^2$	$(0.110)^2\ \bar{Z}_{\delta_e}\bar{Z}_\alpha$	$(0.110)^2\ \bar{M}_\alpha\bar{Z}_\alpha$	$(0.04)^2\ \bar{M}_{\delta_e}\bar{Z}_\alpha$	$(0.12)^2\ \bar{M}_{\dot{\theta}}\bar{Z}_\alpha$
	$(0.110)^2\ \bar{Z}_\alpha\bar{Z}_{\delta_e}$	$(0.150)^2\ \bar{Z}_{\delta_e}^2$	$(0.04)^2\ \bar{M}_\alpha\bar{Z}_{\delta_e}$	$(0.110)^2\ \bar{M}_{\delta_e}\bar{Z}_{\delta_e}$	$(0.110)^2\ \bar{M}_{\dot{\theta}}\bar{Z}_{\delta_e}$
	$(0.110)^2\ \bar{Z}_\alpha\bar{M}_\alpha$	$(0.04)^2\ \bar{Z}_{\delta_e}\bar{M}_\alpha$	$(0.12)^2\ \bar{M}_\alpha^2$	$(0.06)^2\ \bar{M}_{\delta_e}\bar{M}_\alpha$	$(0.114)^2\ \bar{M}_{\dot{\theta}}\bar{M}_\alpha$
	$(0.04)^2\ \bar{Z}_\alpha\bar{M}_{\delta_e}$	$(0.110)^2\ \bar{Z}_{\delta_e}\bar{M}_{\delta_e}$	$(0.06)^2\ \bar{M}_\alpha\bar{M}_{\delta_e}$	$(0.12)^2\ \bar{M}_{\delta_e}^2$	$(0.06)^2\ \bar{M}_{\dot{\theta}}\bar{M}_{\delta_e}$
	$(0.12)^2\ \bar{Z}_\alpha\bar{M}_{\dot{\theta}}$	$(0.110)^2\ \bar{Z}_{\delta_e}\bar{M}_{\dot{\theta}}$	$(0.114)^2\ \bar{M}_\alpha\bar{M}_{\dot{\theta}}$	$(0.06)^2\ \bar{M}_{\delta_e}\bar{M}_{\dot{\theta}}$	$(0.14)^2\ M_{\dot{\theta}}^2$

†obtained from Reference [44]

$$\psi = -4\ V\ Z_{\delta_e} \tag{5.18}$$

where V is the velocity of the aircraft in ft./sec.

The bending frequencies are assumed to be normally distributed with the following means and variances:

$$\bar{\omega}_{b_1} = 30 \quad \text{rad./sec.}$$

$$\bar{\omega}_{b_2} = 50 \quad \text{rad./sec.}$$

$$\overline{(\omega_{b_1} - \bar{\omega}_{b_1})^2} = (0.1)^2 \cdot \overline{\omega_{b_1}^2}$$

$$\overline{(\omega_{b_2} - \bar{\omega}_{b_2})^2} = (0.1)^2 \cdot \overline{\omega_{b_2}^2}$$

The damping ratios of both bending modes are equal with:

$$\zeta_{b_1} = \zeta_{b_2} = 0.01$$

The effect of the bending motion on the measured pitch angle and pitch rate is given by:

$$\theta_i = \theta + \lambda_{b_1} \eta_1 + \lambda_{b_2} \eta_2 \tag{5.19}$$

and

$$\dot{\theta} = \dot{\theta} + \lambda_{b_1} \dot{\eta}_1 \quad \lambda_{b_2} \dot{\eta}_2 \tag{5.20}$$

where λ_{b_1} and λ_{b_2} are the slopes of the bending modes at the location of the gyros, given by:

$$\lambda_{b_1} = 0.025 \quad \text{rad./ft.}$$

$$\lambda_{b_2} = -0.040 \quad \text{rad./ft.}$$

The transfer function, describing the pitch response of the rigid vehicle to an incremental deflection of the elevator, can be obtained from Equation (5.15) as:

$$\frac{\theta(s)}{\delta_e(s)} = \frac{M_{\delta_e} \, [s - Z_\alpha + M_\alpha Z_{\delta_e}/M_{\delta_e}]}{s[s^2 - (Z_\alpha + M_{\dot\theta}) \, s + (Z_\alpha M_{\dot\theta} - M_\alpha)]} \tag{5.21}$$

The transfer function of the actuator is given by:

$$\frac{\delta_e(s)}{\delta_{e_c}(s)} = \frac{20}{s + 20} \tag{5.21}$$

and the transfer functions of the bending deflections are obtained from Equation (5.17) as:

$$\frac{\eta_1(s)}{\delta_e(s)} = \frac{\psi}{s^2 + 2\zeta_{b_1} \, \omega_{b_1} s + \omega_{b_1}^2}$$

$$\frac{\eta_1(s)}{\delta_e(s)} \quad \frac{\psi}{s^2 + 2\zeta_{b_2} \, \omega_{b_2} s + \omega_{b_2}^2} \tag{5.22}$$

The design of the attitude control system will be considered for the flight condition corresponding to the following Mach number and dynamic pressure: $M = 1.03$ and $q = 1160$ lbs./ft.^{2} The corresponding mean values of the dimensional coefficients are given by:

$$\bar{Z}_{\alpha} = -2.275 \quad \text{sec.}^{-1}$$

$$\bar{Z}_{\delta_e} = -0.459 \quad \text{sec.}^{-1}$$

$$\bar{M}_{\alpha} = -46.0 \quad \text{sec.}^{-1}$$

$$\bar{M}_{\delta_e} = -45.9 \quad \text{sec.}^{-1}$$

$$\bar{M}_{\dot{\theta}} = -2.275 \quad \text{sec.}^{-1}$$

The covariance matrix of the uncertainties in these coefficients is then easily obtained from Table 5.3 by substitution of these values. The desired response of the system to an input command as sensed by the attitude gyro is described by a fifth order reference model whose transfer function is given by:

$$\hat{G}(s) = \frac{38880}{(s+1.2)(s^2+2(.5)6s+(6)^2)(s^2+2(.3)30s+(30)^2)} \tag{5.23}$$

This is basically a third order model, which describes the dominant behaviour of the desired response. Its pole locations were determined from normalized step responses for third order systems as given in Reference [5]. A second order mode is then added in order to account for the bending motion which is required to be stable and well-damped for two standard deviations of the bending frequencies ω_{b_1} and ω_{b_2} from their respective nominal values. The frequency of this model bending mode is chosen to be equal to the frequency of the first bending mode of the aircraft, but a damping ratio of $\zeta = .3$ is specified. A desired pole location of the second bending mode is not included in the model, since the contribution of this mode to the response is likely to be insignificant as long as it is stable. The weighting matrix $\underline{Q}$ is easily determined from the coefficients of the model as before.

A block diagram of the system is shown in Figure 5.14. Both pitch angle and pitch rate are fed back, but additional compensation is required to stabilize the system. This may be verified from Figure 5.15 which shows the root-locus for this system for an equal gain of the two feedback signals but without any compensation. The first bending mode is seen to be marginally stable or unstable in this case. In the interest of simplicity it was decided to use a lag filter in an attempt to stabilize this mode. The root-locus departure angles of the first bending poles cannot be changed in this manner, however, without affecting the departure angle of the poles of the second bending mode. For this reason it was found necessary to use two second order lags in order to stabilize both bending modes simultaneously. The transfer function of this filter is given by:

$$G_f(s) = \frac{\omega_{f_1}^2 \; \omega_{f_2}^2}{(s^2+2(.7)\omega_{f_1} s+\omega_{f_1}^2)(s^2+2(.25)\omega_{f_2} s + \omega_{f_2}^2)} \tag{5.24}$$

where the damping ratios have been chosen, leaving the natural frequencies as free design parameters. A lead-lag is also added in order to counteract the adverse effect of the bending filters on the second order rigid vehicle mode. The pole and zero locations of the lead-lag are both designated as free design parameters. Thus, six free design parameters must be selected by the design process:

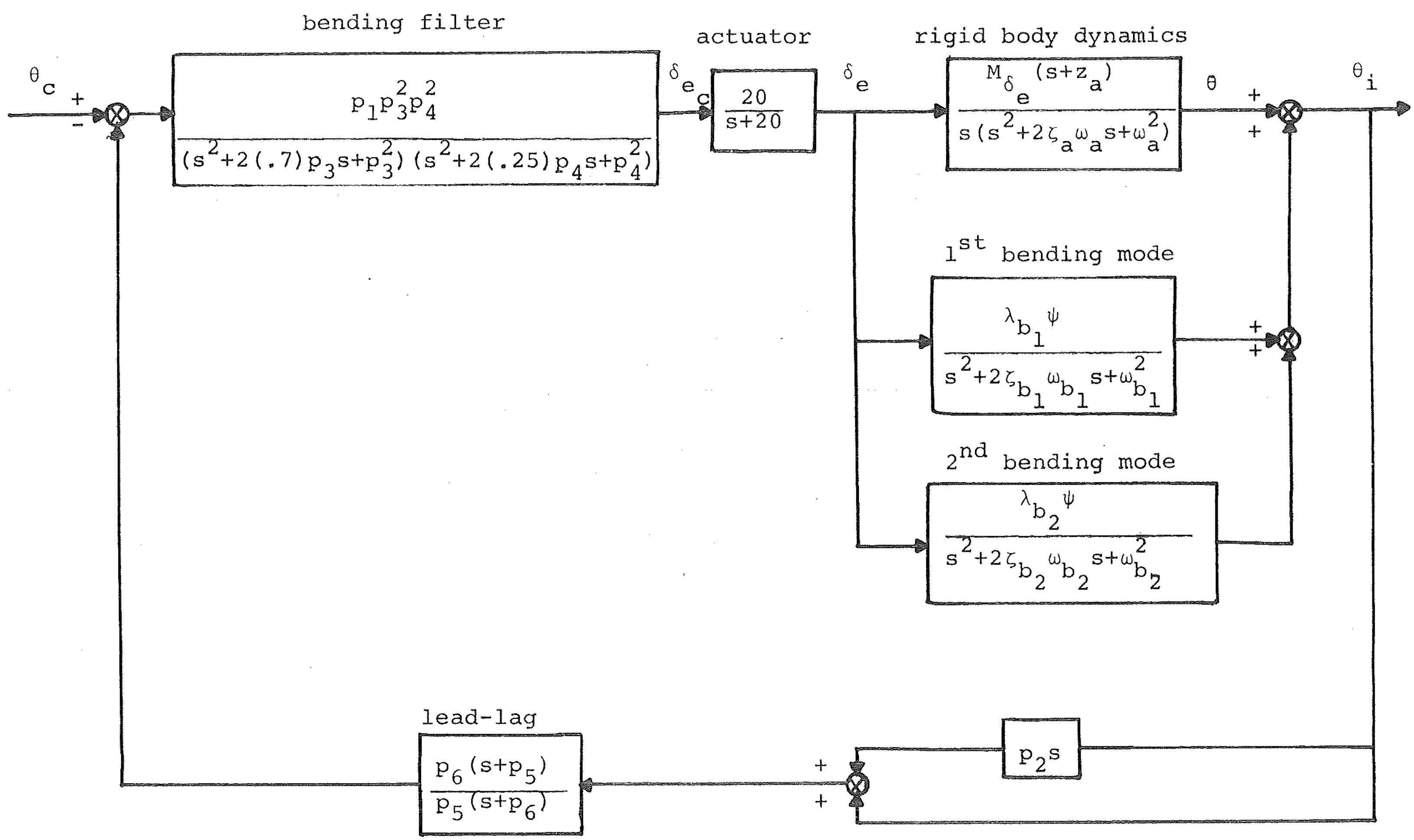


Figure 5.14. Block diagram of aircraft attitude control system.

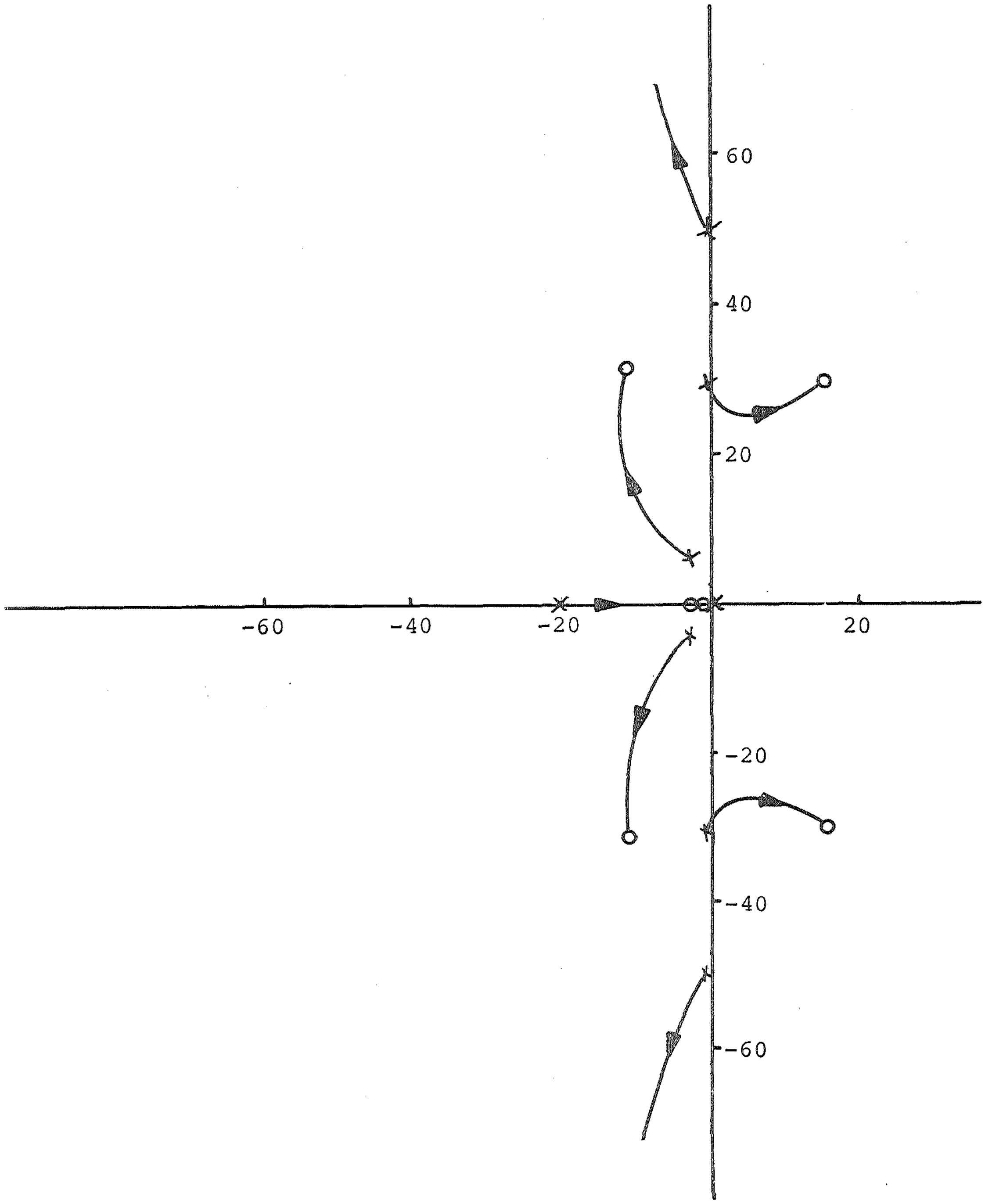


Figure 5.15. Root locus with equal pitch and pitch rate gains but without compensation.

p_1 = static sensitivity of compensation

p_2 = gain of rate feedback

p_3 = first bending filter frequency

p_4 = second bending filter frequency

p_5 = zero location of lead-lag

p_6 = pole location of lead-lag

From Figure 5.14 the complete system may be observed to have 13 poles and 6 zeros.

5.5.1 Sensitivity to Variations of ω_{b_1} and ω_{b_2}

First the nominal design was obtained by minimizing the nominal value of the performance index with respect to the free design parameters setting $\varepsilon = 0$. The value of the sensitivity index due to the uncertainties in the bending mode frequencies was also computed using the following covariance matrix:

$$\underline{R} = \begin{bmatrix} .01\,\overline{\omega_{b_1}^2} & 0 \\ 0 & .01\overline{\omega_{b_2}^2} \end{bmatrix}$$

since the uncertainties in ω_{b_1} and ω_{b_2} are uncorrelated. The resulting parameter and index values are given in Table 5.4 as design no. 1. One of the interesting aspects of this solution is the low value which is chosen for p_3, the natural frequency of the first bending filter mode. This value is actually smaller than the natural frequency of the rigid body mode. The frequency of the second bending

	design number	
parameter	1	2
ε	0	1
p_1	.617	.563
p_2	.155	.155
p_3	2.57	2.41
p_4	38.4	15.7
p_5	6.29	4.16
p_6	27.2	22.0
J_*	.297	.380
J_s	1.44	.034
$\bar{J}$	1.737	.414
μ		.94

Table 5.4 Values of free design parameters and performance indices with uncertainties in both ω_{b_1} and ω_{b_2}

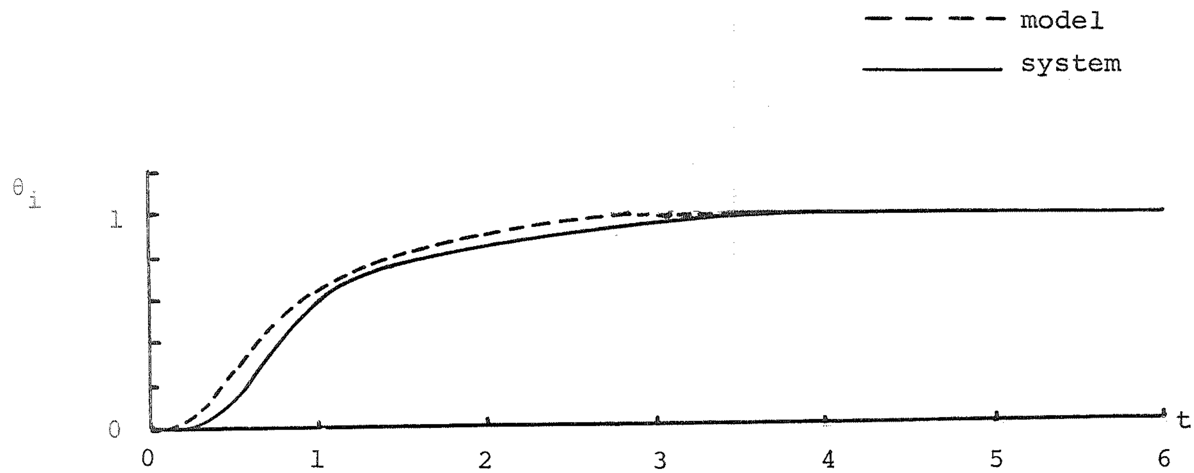
filter mode, p_4, is selected inbetween the bending frequencies.

The solution which is obtained by minimizing $\bar{J}$ for $\varepsilon=1$ is referred to as design no. 2. The most significant effect on the free design parameters, when compared with design no. 1, is that the frequency of the second bending filter has been decreased by more than one half. Other changes are relatively minor with a slight decrease in static sensitivity and an increased amount of lead-lag as shown by the increase in the ratio of p_6 to p_5. The effect of these changes on the sensitivity index is very significant, however, reducing its value by a factor of 40. The trade-off between J_* and

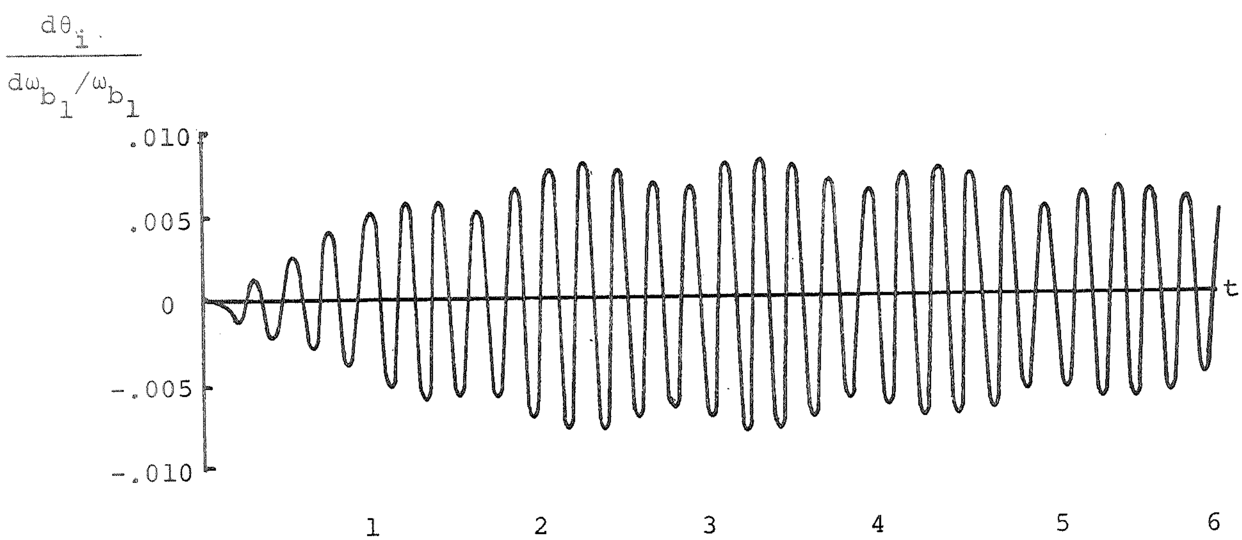
J_s is quite favourable as indicated by the high value of μ.

The normalized step response of design no. 1 is shown in Figure 5.16. The pitch angle response is seen to approximate the model response reasonably well, although the system response is somewhat slower. A similar agreement is observed for the pitch rate response, which indicates a time delaying effect in the system response. No bending motion can be discerned in the pitch and pitch rate responses. The second and third derivatives of the pitch angle show the effect of the bending motion very clearly, however, as lightly damped high frequency oscillations which can be traced to the first bending mode. No signs of the second bending mode can be observed. The sensitivity functions in Figure 5.16 are also a good indicator of the low damping characteristics of the first bending mode. The low damping of the structural bending motion is likely to be very undesirable from the pilot's point of view and may also affect the fatigue life of the structure in the long run. It was found that $\pm 20\%$ changes in the first bending frequency did not produce any extraordinary changes in the bending response or result in additional stability problems.

The step response of design no. 2 is given in Figure 5.17 which shows that the reduction of the sensitivity index has had a major effect on the sensitivity of the system to changes in the first bending mode frequency. This can be observed by comparing the sensitivity function for designs no. 1 and no. 2 which show that the amplitudes for design no. 2 are significantly smaller. The reduction in the system's sensitivity to changes in ω_{b_1} was achieved by reducing the natural frequency of the second bending filter mode from 38.4 rad./sec. to 15.7 rad./sec. This has a significant effect on the first bending mode without affecting the stability of the second bending mode.

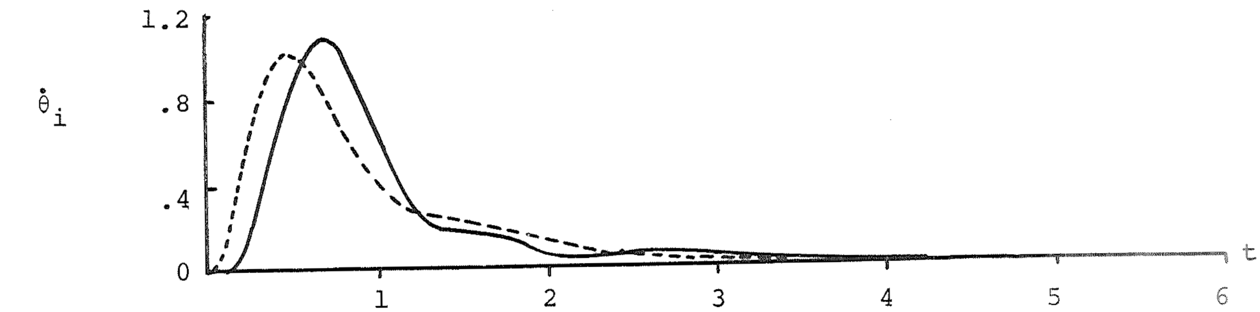


Pitch response at gyro station, $\underline{\xi} = \underline{\xi}_*$

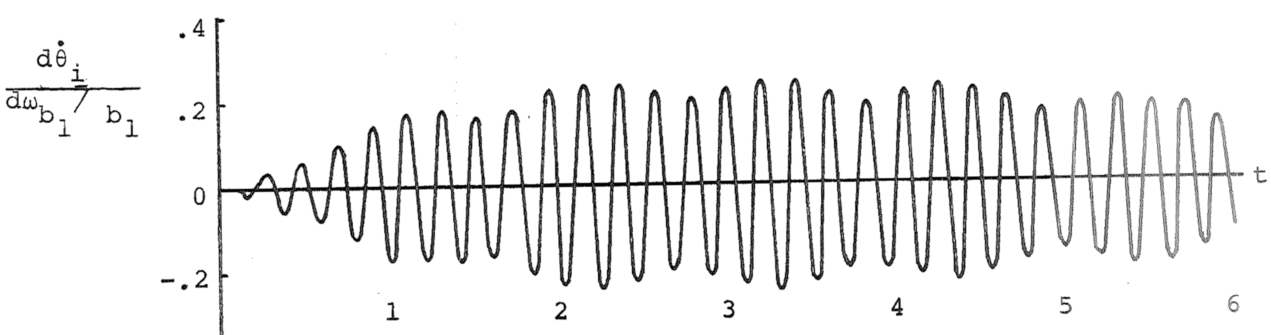


Sensitivity of pitch response w.r.t. ω_{b_1}

Figure 5.16. Step response, design no. 1. (continued)



Pitch rate response at gyro station, $\underline{\xi}=\underline{\xi}_*$



Sensitivity of pitch rate response w.r.t. ω_{b_1}

Figure 5.16. Step response, design no. 1. (continued)

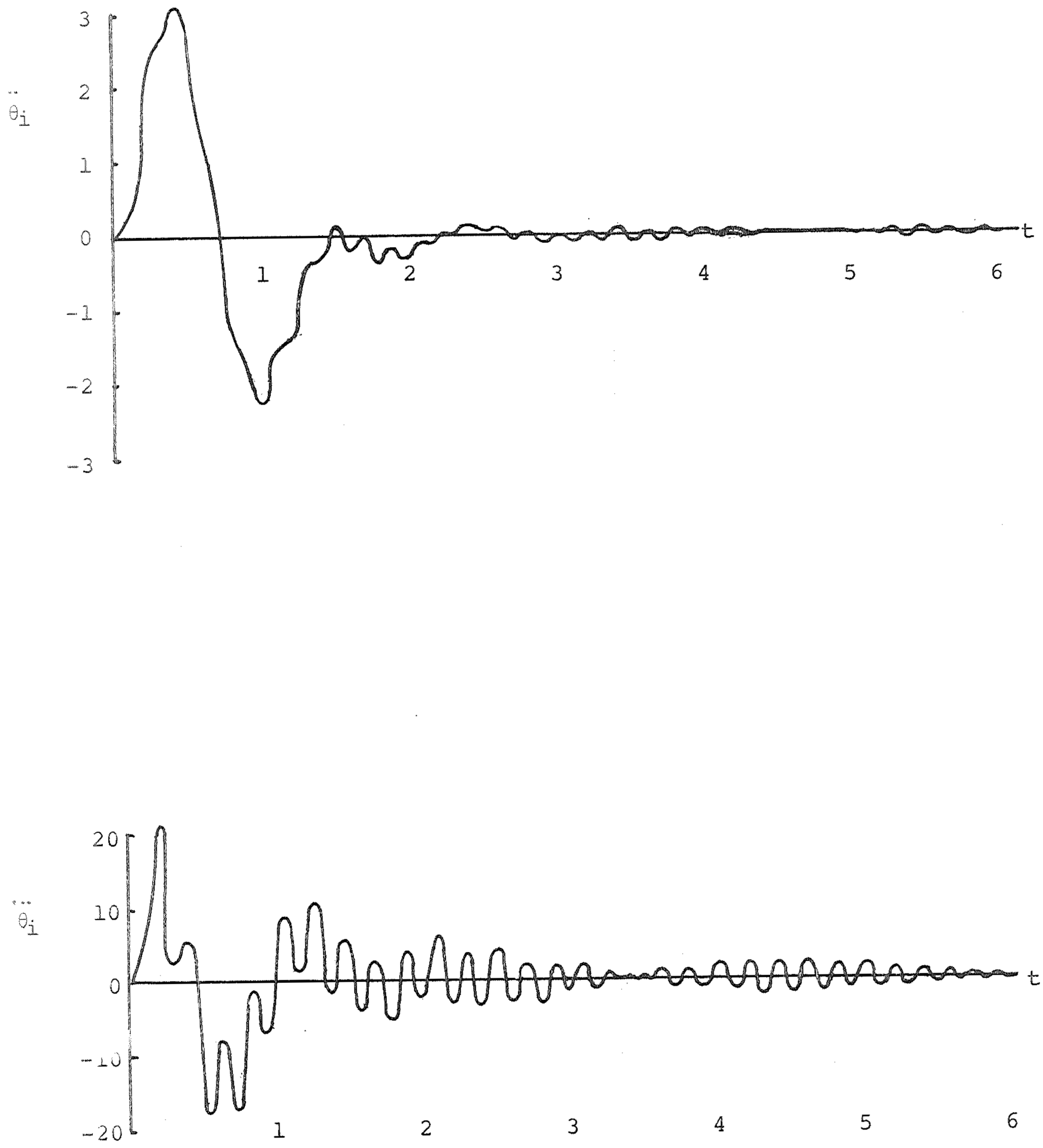
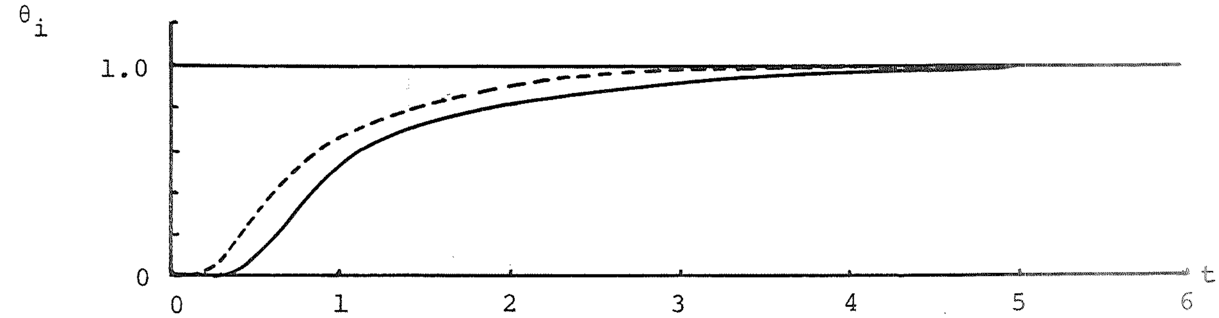
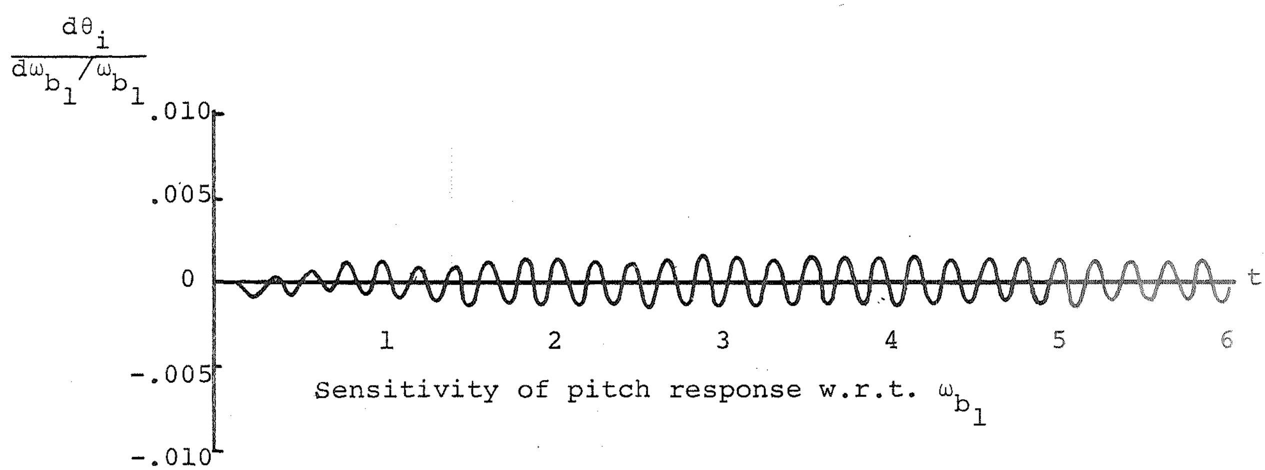


Figure 5.16. Step response, design no. 1.

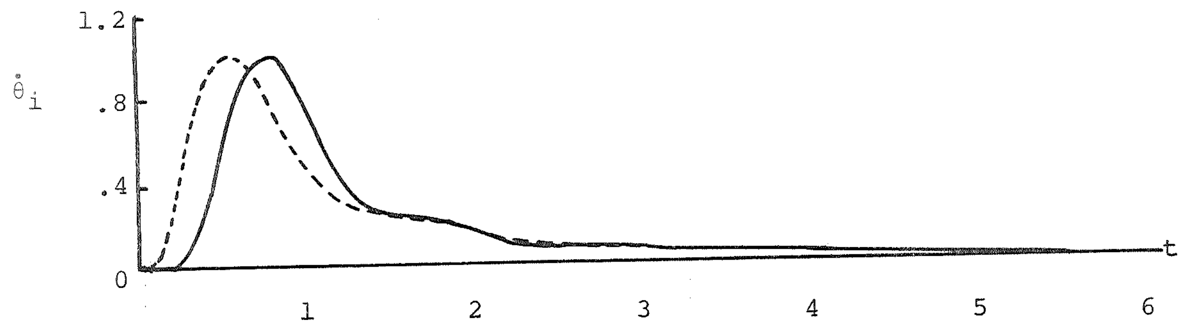


Pitch response at gyro station, $\underline{\xi}=\underline{\xi}_*$

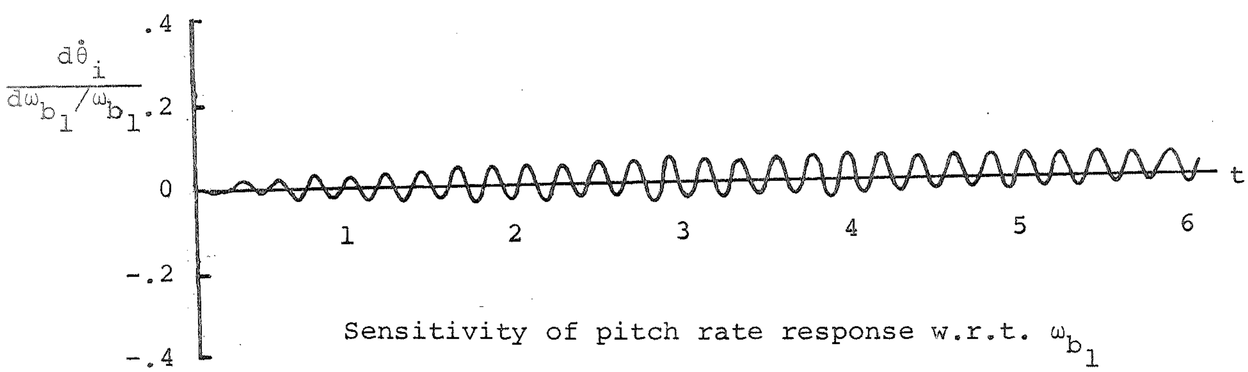


Sensitivity of pitch response w.r.t. ω_{b_1}

Figure 5.17. Step response, design no. 2. (continued)

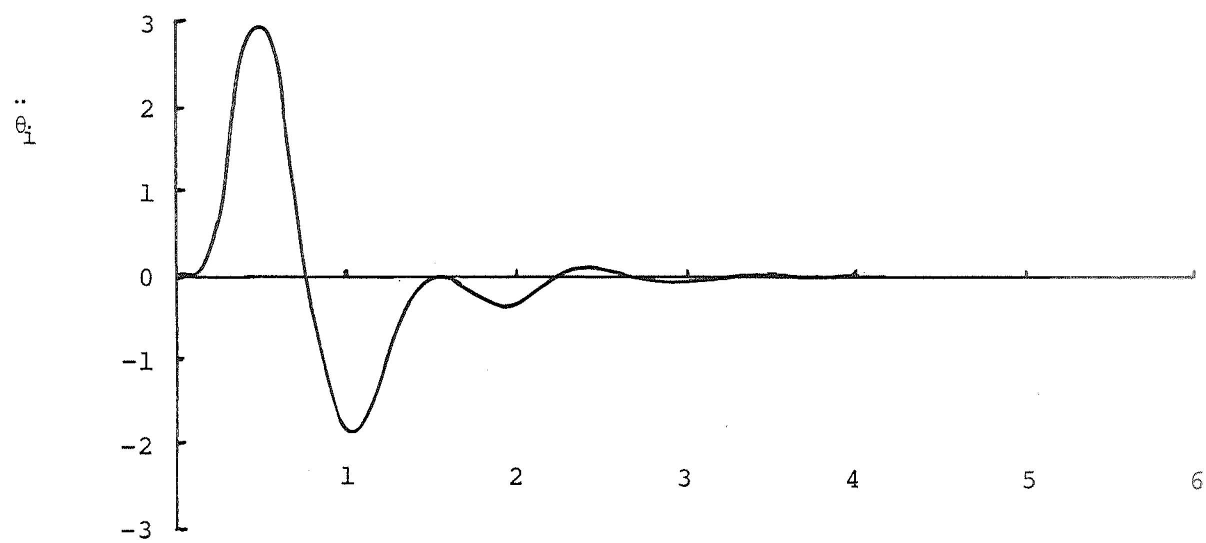


Pitch rate response at gyro station, $\underline{\xi}=\underline{\xi}_*$



Sensitivity of pitch rate response w.r.t. ω_{b_1}

Figure 5.17. Step response, design no. 2.(continued)



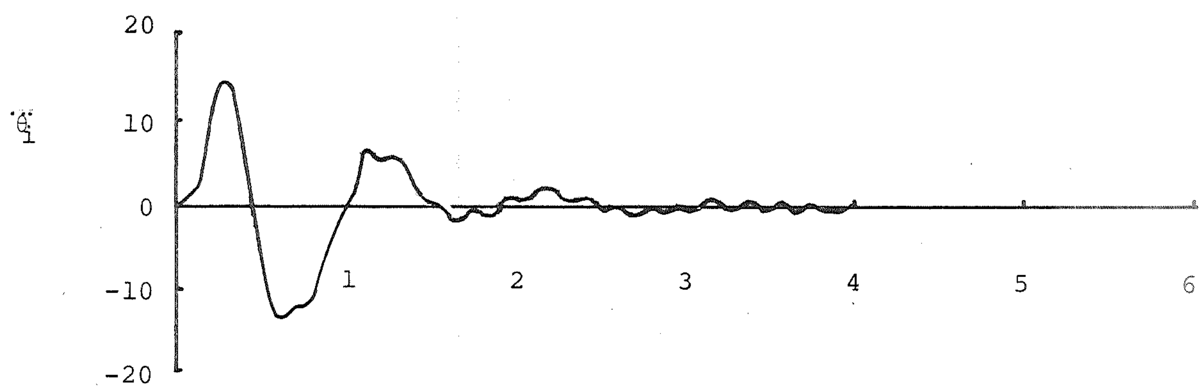


Figure 5.17. Step response, design no. 2.

Comparison of the step responses for the two designs in Figures 5.16 and 5.17 shows that the initial part of the pitch angle response has been slightly slowed down by the effort to reduce the sensitivity of the system. The change in the settling time is insignificant, however, being less than half a second. The important improvement in the response may be observed from the second and third derivatives of the pitch response. The effect of the bending motion on the second derivative has virtually disappeared and is very small in the case of the third derivative. The structural response of design no. 2 is, therefore, preferable by far to the response of design no. 1.

5.5.2 Sensitivity to Variation of Dimensional Coefficients

The uncertainties in the knowledge of the dimensional coefficients is expressed by the covariance matrix in Table 5.3. The corresponding sensitivity index for design no. 1 was found to be:

$$J_s = 0.03$$

This is a very small value in comparison to the sensitivity index which was obtained for this design with respect to the bending frequencies. No reduction of the sensitivity index was achieved by minimization of the expected value of the performance index. It is therefore concluded that design no. 1 corresponds approximately to the minimum value of J_s in this case, since otherwise it would be possible to obtain some reduction in its value.

Figure 5.18 shows the sensitivity functions of the output response of design no. 1 with respect to each of the five dimensional coefficients. The off-nominal response of the system is also given in Figure 5.19 for two standard deviations of M_{δ_e}, which is the most critical of these parameters. Since all the dimensional coefficients

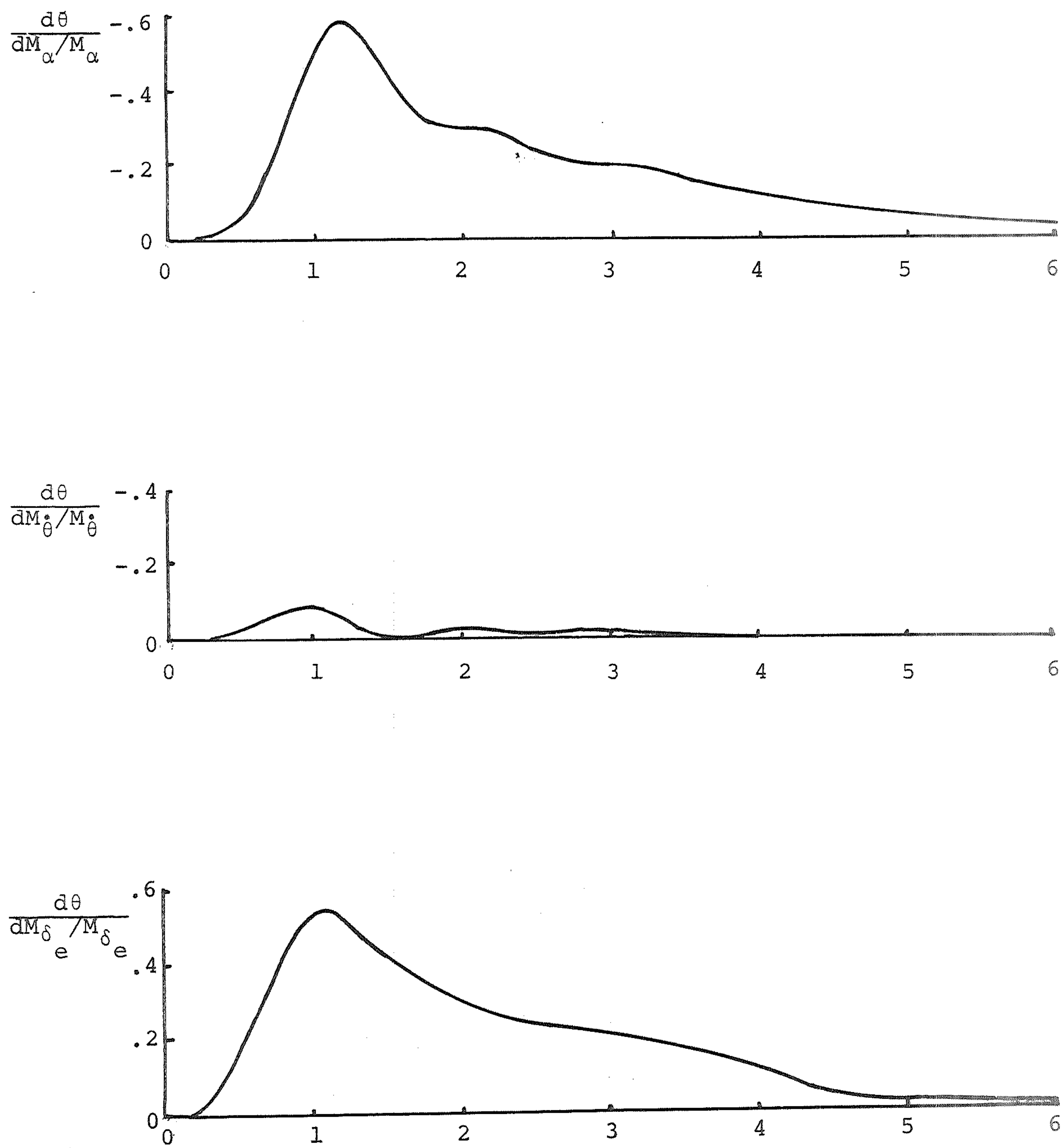


Figure 5.18. Sensitivity of pitch response w.r.t. dimensional coefficients

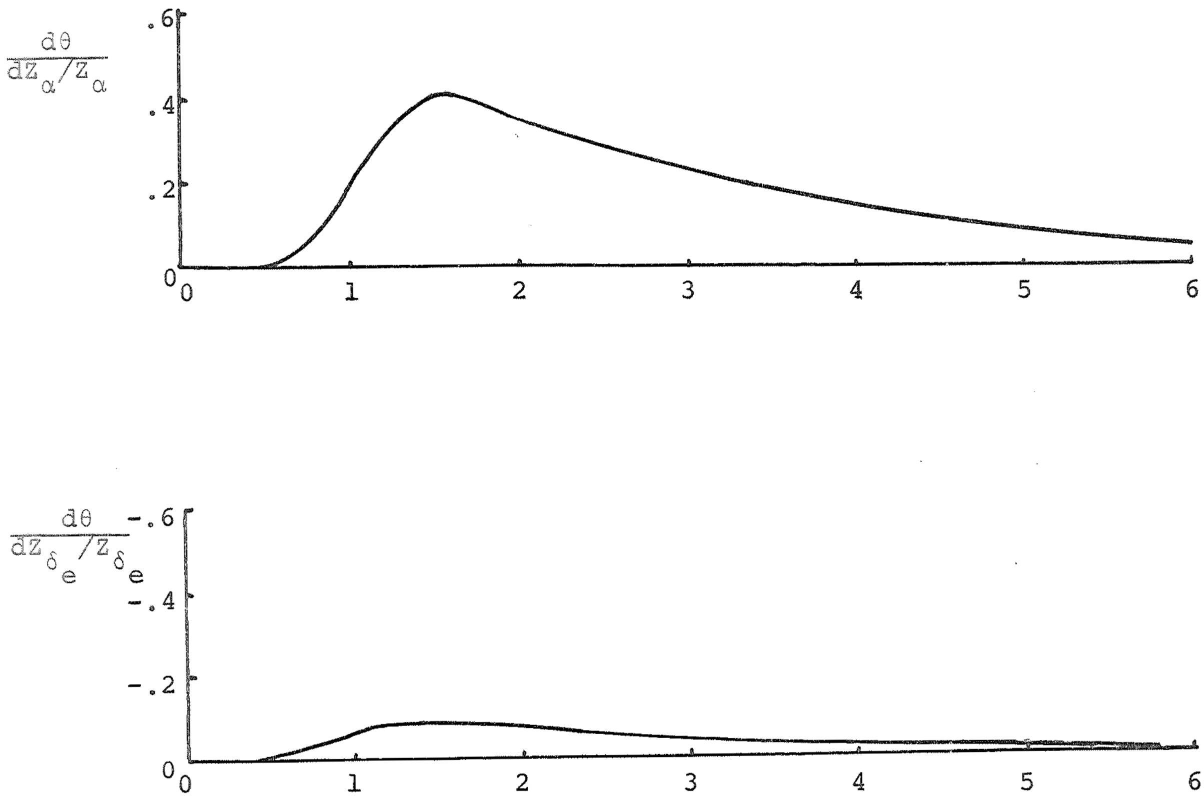


Figure 5.18. Sensitivity of pitch response with w.r.t. dimensional coefficients (continued)

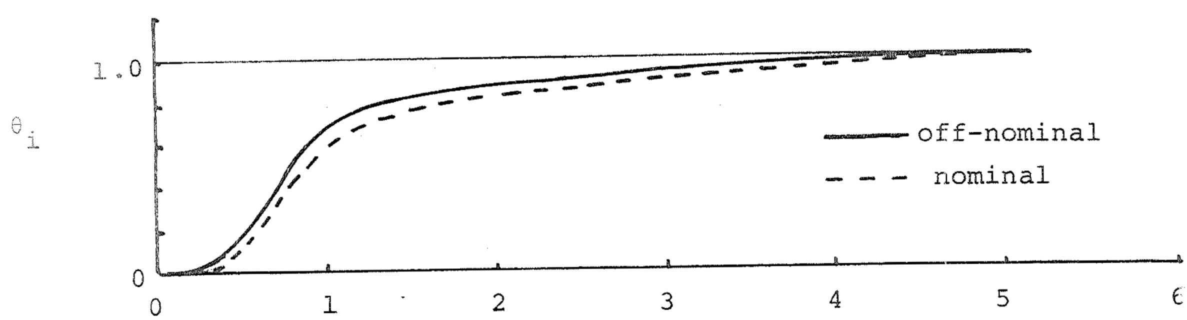


Figure 5.19. Off-nominal pitch response, design no. 1

$M_{\delta_e} = 1.24\ M^*_{\delta_e}$

are correlated it would be unrealistic to vary only one of them independently. The response in Figure 5.19 is therefore computed by using the conditional means of the four remaining coefficients given the specified variation of M_{δ_e}.

5.6 Discussion of Results

The first design example shows that the sensitivity design method developed in Chapter 3 can be used effectively in order to reduce the effect of parameter uncertainties. Moreover, it may be done in a way which is consistent with meeting realistic response requirements. Thus, by varying the weighting of the sensitivity index it has been found possible to control the sensitivity of this system with relatively minor changes in the nominal response. This property will of course vary from system to system as indicated by the μ trade-off parameter.

The second design example differs from the previous one in that the specified uncertainties of the design parameters were not found to have a critical effect on the response of the system. The uncertainties of the structural bending frequencies were, however, found to have a significant effect on the value of the sensitivity index. This can be attributed to the light damping of the first bending mode. Thus, a moderate variation of that mode can have a large cumulative effect when integrated over a long time period. Reduction in the sensitivity of this bending mode was achieved by increasing the effect of the bending filter at the first bending frequency. The sensitivity index, therefore, could be used to detect the low damping of the structural response. The reduction of the value of this index was found to be an effective tool for suppressing the undesirable excitation of this response.

The low sensitivity of the nominal design to uncertainties in the dimensional coefficients was reflected by the small contribution of the sensitivity index to the expected value of the performance index.

APPENDIX A

Derivation of Closed-Loop Pole Sensitivities

In order to derive an expression for the derivative of a closed-loop system pole with respect to an open-loop parameter, some relationship between the open-loop and closed-loop system characteristics must be used. A particularly convenient relationship of this type is given by:

$$1 + G_{OL} = \frac{\prod_{i=1}^{n} (s - p_i)}{\prod_{j=1}^{n} (s - \tilde{p}_j)} \tag{A-1}$$

where G_{OL} is the total open-loop transfer function, and p_i and $\tilde{p}_j$ are the closed and open-loop poles respectively, of a single-loop system. The validity of Equation (A-1) should be clear from the well-known fact that the zeros of $1 + G_{OL}$ are the closed-loop system poles and the poles of $1 + G_{OL}$ must be the same as the poles of G_{OL} itself. The system is also assumed to have at least one more open-loop pole than zeros.

Taking the derivative with respect to an open-loop parameter, ξ, on both sides of Equation (A-1) gives:

$$\frac{\partial G_{OL}}{\partial \xi} = \frac{\prod_{i=1}^{n} (s-p_i)}{\prod_{j=1}^{n} (s-\tilde{p}_j)} \left[\sum_{j=1}^{n} \frac{\frac{\partial \tilde{p}_j}{\partial \xi}}{(s-\tilde{p}_j)} - \sum_{i=1}^{n} \frac{\frac{\partial p_i}{\partial \xi}}{(s-p_i)} \right] \tag{A-2}$$

where it is assumed that all the open and closed-loop poles are distinct. Using Equation (A-1) this equation can be written:

$$\sum_{i=1}^{n} \frac{\frac{\partial p_i}{\partial \xi}}{(s-p_i)} - \sum_{j=1}^{n} \frac{\frac{\partial \tilde{p}_j}{\partial \xi}}{(s-\tilde{p}_j)} = - \frac{\frac{\partial G_{OL}}{\partial \xi}}{1 + G_{OL}} \tag{A-3}$$

Multiplying both sides of this equation by $(s-p_k)$ and setting $s = p_k$ gives the following result:

$$\frac{\partial p_k}{\partial \xi} = \left[\frac{(s - p_k)\,\dfrac{\partial G_{OL}}{\partial \xi}}{1 + G_{OL}}\right]_{s = p_k} \tag{A-4}$$

where only one term due to the left hand side of Equation (A-3) remains. The rest of the terms all become zero when $s = p_k$, since p_k is distinct from the remaining open and closed-loop poles.

S_{OL}, $\tilde{p}_j$, and $\tilde{z}_j$ can now be substituted for ξ in order to find the sensitivities of the $k^{\underline{th}}$ closed-loop pole to changes in these parameters:

$$\frac{\partial p_k}{\partial S_{OL}} = -\left[\frac{(s - p_k)\,\dfrac{\partial G_{OL}}{\partial S_{OL}}}{1 + G_{OL}}\right]_{s = p_k} = -\frac{1}{S_{OL}}\left[\frac{(s - p_k)\,G_{OL}}{1 + G_{OL}}\right]_{s = p_k} \tag{A-5}$$

which is obtained by substituting the following relation:

$$\frac{\partial G_{OL}}{\partial S_{OL}} = \frac{\partial}{\partial S_{OL}}\left[\frac{S_{OL}\prod_{i=1}^{m}(1 - \frac{s}{\tilde{z}_i})}{\prod_{j=1}^{n}(1 - \frac{s}{\tilde{p}_j})}\right] = \frac{G_{OL}}{S_{OL}} \tag{A-6}$$

The sensitivity of p_k with respect to S_{OL} is then expressed by:

$$S^k_{S_{OL}} = \frac{\partial p_k}{\partial S_{OL}/S_{OL}} = -\left[\frac{(s - p_k)\,G_{OL}}{1 + G_{OL}}\right]_{s = p_k} \tag{A-7}$$

Substituting $\tilde{p}_j$ for ξ in Equation (A-4) gives:

$$S_{\tilde{p}_j}^{k} = \frac{\partial \tilde{p}_k}{\partial \tilde{p}_j} = -\left[\frac{(s - p_k)\dfrac{\partial G_{OL}}{\partial \tilde{p}_j}}{1 + G_{OL}}\right]_{s=p_k} = -\left[\frac{s(s - p_k)}{\tilde{p}_j(s - \tilde{p}_j)} \frac{G_{OL}}{1 + G_{OL}}\right]_{s=p_k} \tag{A-8}$$

where the following expression has been sutstituted for the derivative of G_{OL}:

$$\frac{\partial G_{OL}}{\partial \tilde{p}_j} = \frac{s}{\tilde{p}_j} \frac{G_{OL}}{(s - \tilde{p}_j)} \tag{A-9}$$

Substituting the expression for $S_{S_{OL}}^{k}$ in Equation (A-8) then yields:

$$S_{\tilde{p}_j}^{k} = \frac{p_k}{\tilde{p}_j} \frac{S_{S_{OL}}}{(p_k - \tilde{p}_j)} \tag{A-10}$$

The sensitivity of p_k to the open-loop zeros is similarly obtained by replacing ξ by $\tilde{z}_j$, which gives the following expression for $S_{\tilde{z}_j}^{k}$:

$$S_{\tilde{z}_j}^{k} = \frac{p_k}{\tilde{z}_j} \frac{S_{S_{OL}}^{k}}{(\tilde{z}_j - p_k)} \tag{A-11}$$

APPENDIX B

Calculation of Derivative Matrices

In order to compute the derivative matrices of Section 4.3 it is sufficient to obtain derivatives of the following form:

$$\frac{\partial q}{\partial p_i} \quad , \quad \frac{\partial q}{\partial \xi_i} \quad , \quad \frac{\partial^2 q}{\partial p_i \partial \xi_j}$$

where q is a scalar function of the design parameter vectors $\underline{p}$ and $\underline{\xi}$:

$$q = q\ (\ \underline{p}\ ,\ \underline{\xi}\ ) \tag{B-1}$$

For a given set of values of the design parameters, the first derivative of q with respect to p_i can be approximated by:

$$\frac{\partial q}{\partial p_i} \cong \frac{q(p_i + \Delta p_i) - q(p_i - \Delta p_i)}{2\Delta p_i} \tag{B-2}$$

where all the parameters are held constant except for p_i, whose increment, Δp_i, is some fraction of its nominal value as an example. Similarly:

$$\frac{\partial q}{\partial \xi_i} \cong \frac{q(\xi_i + \Delta\xi_i) - q(\xi_i - \Delta\xi_i)}{2\Delta\xi_i} \tag{B-3}$$

An approximate expression for the second derivative of q with respect to p_i and ξ_j can then be obtained as follows:

$$\frac{\partial}{\partial p_i}\left[\frac{\partial q}{\partial \xi_j}\right] \cong \frac{\frac{\partial q}{\partial \xi_j}(p_i + \Delta p_i) - \frac{\partial q}{\partial \xi_j}(p_i - \Delta p_i)}{2\Delta p_i} \tag{B-4}$$

Equation (B-3) is then used to evaluate the first derivatives in Equation (B-4) which gives:

$$\frac{\partial^2 q}{\partial p_i \partial \xi_j} \cong$$

$$\frac{q(p_i+\Delta p_i,\xi_j+\Delta\xi_j) - q(p_i+\Delta p_i,\xi_j-\Delta\xi_j) - q(p_i-\Delta p_i,\xi_j+\Delta\xi_j) + q(p_i-\Delta p_i,\xi_j-\Delta\xi_j)}{4\Delta p_i \Delta\xi_j} \tag{B-5}$$

In order to compute these three derivatives of q with respect to two of the design parameters it is, therefore, necessary to evaluate eight different values of q. This number can be reduced to four by using the following approximation:

$$q(p_i+\Delta p_i) \cong \frac{q(p_i+\Delta p_i,\xi_j+\Delta\xi_j) + q(p_i+\Delta p_i,\xi_j-\Delta\xi_j)}{2} \tag{B-6}$$

This result can then be substituted into Equation (B-2) to give:

$$\frac{\partial q}{\partial p_i} \cong$$

$$\frac{q(p_i+\Delta p_i,\xi_j+\Delta\xi_j) + q(p_i+\Delta p_i,\xi_j-\Delta\xi_j) - q(p_i-\Delta p_i,\xi_j+\Delta\xi_j) - q(p_i-\Delta p_i,\xi_j-\Delta\xi_j)}{4\Delta p_i} \tag{B-7}$$

An analogous expression for the derivative of q with respect to ξ_j is obtained by interchanging p_i and ξ_j.

Clearly, Equations (B-5) and (B-7) require the same values of q, which only needs to be evaluated four times. The accuracy of the first derivatives, as expressed by Equation (B-7), is less than that obtained by using Equation (B-2) because of the error introduced by Equation (B-6). This deterioration in numerical accuracy must be weighed against the reduction in the computation of q, which can be important when q is a complicated function of the parameters. When q is an element of an n-dimensional vector, these derivatives must be found for all possible combinations of the vector components and the design parameters.

If $\underline{p}$ is a k-dimensional vector and $\underline{\xi}$ is ℓ-dimensional this means that $n\cdot(k+\ell)$ first derivatives and $n\cdot k\cdot \ell$ second derivatives have to be computed. Hence, the savings achieved by using Equation (B-7) become relatively smaller as the number of design parameters increases. This is reflected by the ratio of the number of evaluations needed for determining the first and second derivatives when these are calculated separately. This ratio expresses the additional effort required for computing the first derivatives separately and is given by:

$$p = \frac{2\cdot n\cdot (k+\ell)}{4\cdot n\cdot k\cdot \ell} = \frac{1}{2}\left[\frac{1}{\ell} + \frac{1}{k}\right] \tag{B-8}$$

since two values are needed for each first derivative and four for each second derivative. For large values of ℓ and k this ratio obviously becomes small.

These approximations have been found to be relatively accurate in calculating the derivative matrices of the closed-loop characteristic coefficients and the initial condition vector. The increments of the design parameters have been chosen to be 5-10% of the current value of these parameters.

APPENDIX C

Computer Programs

The computer programs which are listed in this appendix can be used to determine the minimum of the expected value of a quadratic performance index with respect to the specified free design parameters. These programs consist of a main program and eight subroutines in addition to utilizing five standard subroutines from the IBM System/360 Scientific Subroutine Package. The numerical techniques which are used were described in detail in Chapter 4. The basic function of each program is explained by comment cards but some additional information about these programs must be given.

The MAIN program was developed from a program written by Rediess [31] for minimizing a quadratic performance index for known, deterministic design parameters. Only the basic structure of the original program has been retained taking advantage of the computational techniques which were derived in Chapter 4. All the input data required is read by the main program using the following input cards:

Card no. 1: FORMAT (3I4, F8.4, I8, 3F8.4, I4)
N M K STEP ITMAX H DIL EPS KK

Card no. 2: FORMAT (5E20.8)
PAR(1) . . . PAR(K)

Card no. 3: FORMAT (5E20.8)
ALPHA(1) . . . ALPHA (N)

Card no. 4: FORMAT (5E20.8)
R(1) . . . R(KK*KK)

Card no. 5: FORMAT (5E20.8)
SI(1) . . . SI(KK)

Explanation of the input parameters:

N = order of the system

M = number of zeros in system transfer function

STEP = step size in the direction of the gradient (defined in Section 4.4)

ITMAX = maximum number of iterations of the gradient procedure

H = minimum improvement in performance index for continuing the gradient search, expressed as a fraction of the performance index value

DIL = minimum difference between the predicted and actual changes in the performance index, specified as a fraction of the change predicted by the first order approximation

EPS = weighting coefficient multiplying the sensitivity index

KK = number of variable design parameters

PAR(I) = $i^{\underline{th}}$ free design parameter

ALPHA(I) = $i^{\underline{th}}$ characteristic coefficient of the model in the system's n-dimensional space

R(I) = $i^{\underline{th}}$ element of the variable parameter covariance matrix in the general storage mode

SI(I) = $i^{\underline{th}}$ variable design parameter.

Note that more than one card may be required for the input parameters on cards no. 2 to no. 5, since only five values can be put on each card as indicated by the FORMAT statements.

Much of the MAIN program output is self-explanatory except for the following:

DAP, DXP, DAS, DXS, DAPS, DXPS

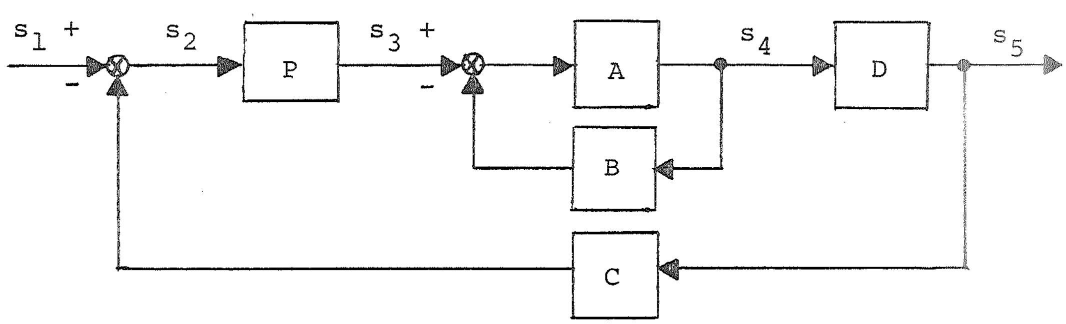
which are first derivatives and cross derivatives of $\underline{a}$ and $\underline{x}_0$ with respect to $\underline{p}$ and $\underline{\xi}$ as indicated by the notation. These quantities are only printed for the initial value of $\underline{p}$. The quantities

PI , DPI , TQX , TQDX

are printed for each iteration of the process and represent, respectively, the value of the performance index, the change in this value as a result of the preceding iteration, the nominal performance index and the sensitivity index. Thus:

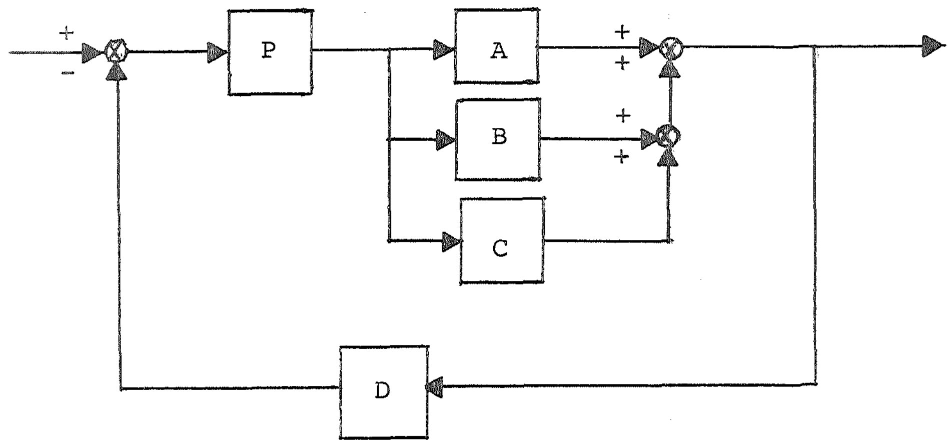
PI = TQX + EPS*TQDX

The function of the MAIN program is to compute the value and the gradient of the performance index and change the values of the free design parameters in an iterative manner such as to minimize the value of the performance index. For this purpose it uses the following subroutines: subroutine SYST which computes all the closed-loop system coefficients, the corresponding initial conditions, and the derivatives of all these quantities with respect to $\underline{p}$ and $\underline{\xi}$. For this purpose it uses the subroutines ROOTIN, SWEEP, and ROOTS for forming the closed-loop transfer function polynomials. In its present form the ROOTIN program has four basic modes, depending on the structure of the block diagram. These modes correspond to the following input/output paths:



mode	input/output
1	$s_2 \to s_3$
2	$s_3 \to s_4$
3	$s_2 \to s_4$
4	$s_1 \to s_5$

A fifth mode which is useful for including structural modes is represented by the following block diagram:



The mode number is specified in the ROOTS subroutine by a statement of the form:

MODE = I

where I is an integer. This subroutine also contains information about the open-loop system roots, which are specified by statements of the following form:

RPPX(I) = ...

CPPX(I) = ...

RPZX(I) = ...

CPZX(I) = ...

The first two characters indicate a real or complex part of the root, the third character distinguishes poles and zeros and the last character identifies the block to which the root belongs. Thus X would be replaced by A, B, C ... etc. The numbers of roots in each block must, furthermore, be specified by statements of the form:

NPX = ...

NZX = ...

representing the number of poles and zeros in X. It should be noted that a complex root pair is counted as a single root. The static sensitivities of the individual open-loop transfer functions are similarly specified by statements of the form:

SX = ...

where the second character identifies the block. These open-loop poles, zeros and static sensitivities can be written as any functions of the free and variable design parameters. An example of the ROOTS program is included in this appendix.

The MAIN program uses the STST and RTRT subroutines to compute the solutions of all the matrix equations which must be solved in order to obtain the value and gradient of the performance index. These subroutines are straightforward mechanizations of the matrix solutions of Chapter 4. Some of the essential computations are made in double precision for increased accuracy. It may be necessary to change the time scale of the system equations in order to prevent an overflow or underflow during the inversion of the polynomial matrix in STST. An overflow in the computation of its determinant indicates that the response should be slowed down whereas an underflow requires a speeding up of the system response. The scaling is achieved by scaling of the roots in the ROOTS subroutine with the static sensitivities unchanged unless the block under consideration contains pure integration or differentiation. The coefficients of the model must of course be scaled by the same amount.

```
C ***********************************************************************
C
C      THORGEIR PALSSON, MIT DEPT. OF AERONAUTICS AND ASTRONAUTICS, 1971
C
C ***********************************************************************
C
C
C      MAIN PROGRAM FOR MINIMIZING THE EXPECTED VALUE OF A QUADRATIC PER-
C      FORMANCE INDEX WITH RESPECT TO SPECIFIED FREE DESIGN PARAMETERS
C      USING AN ITERATIVE GRADIENT ALGORITHM
C
C      DESCRIPTION OF INPUT DATA
C
C      N = ORDER OF THE SYSTEM
C      M = NUMBER OF SYSTEM ZEROS
C      K = NUMBER OF FREE DESIGN PARAMETERS
C      STEP = SIZE OF THE GRADIENT STEP
C      ITMAX = MAXIMUM NUMBER OF ITERATIONS
C      H = MINIMUM PERCENTAGE IMPROVEMENT OF PERFORMANCE INDEX PER ITERATION
C      DIL = MINIMUM PERCENTAGE DIFFERENCE BETWEEN ACTUAL AND PREDICTED
C            CHANGE IN PERFORMANCE INDEX
C      EPS = WEIGHTING COEFFICIENT OF SENSITIVITY INDEX
C      KK = NUMBER OF VARIABLE DESIGN PARAMETERS
C      PAR(I) = INITIAL VALUE OF I-TH FREE DESIGN PARAMETER
C      ALPHA(I) = I-TH CHARACTERISTIC COEFFICIENT OF MODEL
C      R(I) = I-TH ELEMENT OF PARAMETER COVARIANCE MATRIX IN GENERAL
C             STORAGE MODE
C      SI(I) = NOMINAL VALUE OF I-TH VARIABLE DESIGN PARAMETER
C
C      REMARKS
C
C      IF KK=0, R(I) AND SI(I) DO NOT HAVE TO BE SPECIFIED
C
C      IN ITS PRESENT FORM THE PROGRAM CAN BE USED FOR UP TO FIFTEENTH
C      ORDER SYSTEMS WITH TEN FREE DESIGN PARAMETERS AND TEN VARIABLE
C      DESIGN PARAMETERS
```

```
C
C       SUBROUTINES REQUIRED
C
C         INCON                     SNORM
C         ROOTIN                    GMPRD
C         ROOTS                     GMTRA
C         RTRT                      MINV ( DOUBLE PRECISION )
C         STST                      MPROD (DOUBLE PRECISION VERSION OF GMPRD )
C         SWEEP                     PMPY
C         SYST                      PADD
C
C ******************************************************************************
C
C
      DIMENSION ACOF(20),EE(20),DD(20),TZ(20),PZ(20),D(20)
      DIMENSION P1(400),P2(400),PX(400),P(400),PXX(400),PT(400)
      DIMENSION XX(400),XX0(400),Q(400),DX(400)
      DIMENSION X0(20),PAR(10),PAROLD(10),GPI(10),GPIOLD(10),ALPHA(20)
      DIMENSION SI(10),GSTEP(10),DIR(10),PSAVE(10),BCOF(20)
      DIMENSION R(100),SR(200),ST(200),TR(200),TT(200),P1D(200),P2D(200)
      DIMENSION W(400),XW(400),WP(400),U(400),V(400)
      DIMENSION UV(400),PV(20),PX0(400),FF(10),DVP(200)
      DIMENSION P1S(400)
C
      COMMON/CONST/N,M,K,KK,KKK,NN,NNN,NPO,NMO,NNPO,NNMO,NMON,NKK
      COMMON/DERIV/DAP(200),DXP(200),DAS(200),DXS(200),DAPS(2000),DXPS(2
     1000)
C
    1 FORMAT(1H1,48HCONTROL SYSTEM DESIGN VIA PARAMETER OPTIMIZATION)
    3 FORMAT(3I4,F8.4,I8,5F8.4,I4)
  103 FORMAT(1H0,2X,1HN,3X,1HM,3X,1HK,2X,4HSTEP,5X,5HITMAX,4X,1HH,7X,
     13HDIL,5X,3HEPS,5X,2HKK)
  104 FORMAT(1H0,'PARAMETER COVARIANCE MATRIX')
  105 FORMAT(1H0,'NOMINAL PARAMETER VALUES')
  106 FORMAT(1H0,31HSTATE VECTOR WEIGHTING MATRIX Q)
  107 FORMAT(1H0,24HINITIAL PARAMETER VECTOR)
```

```
  109 FORMAT(1HO,30HMODEL DENOMINATOR COEFFICIENTS)
  110 FORMAT(1HO,'DESIGN PARAMETER VALUES')
    5 FORMAT(1HO,27HTERMINATED BECAUSE IT=ITMAX)
    7 FORMAT(1HO,28HSTOPPING CONDITION SATISFIED)
    8 FORMAT(1HO,46HFINAL VALUE OF PARAMETER VECTOR IS GIVEN BELOW)
    9 FORMAT(1HO,'NEGATIVE PERFORMANCE INDEX')
   10 FORMAT(1HO,39HINITIAL CONDITION VECTOR IS GIVEN BELOW)
   11 FORMAT(1HO,16HITERATION NUMBER,I5)
   12 FORMAT(1HO,5HSTEP=,F16.8)
   14 FORMAT(1HO,18HPERFORMANCE INDEX=,E16.8)
   20 FORMAT(1HO,5E20.8)
   22 FORMAT(1HO,5X,24HDENOMINATOR COEFFICIENTS)
   23 FORMAT(1HO,5X,24HINITIAL CONDITION VECTOR)
  123 FORMAT(1HO,5X,22HNUMERATOR COEFFICIENTS)
   24 FORMAT(5E16.8)
  125 FORMAT(1HO,42HTHE GRADIENT OF THE PERFORMANCE INDEX, GPI)
  126 FORMAT(1HO,6HGPIOLD)
  999 FORMAT(1HO,9E12.4)
  997 FORMAT(F10.2,I6)
  995 FORMAT(1HO,'CHARACTERISTIC COEFFICIENTS')
  994 FORMAT(1HO,'INITIAL CONDITION VECTOR')
  993 FORMAT(1HO,14X,'PI',18X,'DPI',17X,'TQX',17X,'TQDX')
  802 FORMAT (1HO,'PARABOLIC STEP')
  803 FORMAT (1HO,'PARABOLIC STEP FAILED')
  804 FORMAT(1HO,'GRADIENT DIRECTION COSINES')
  805 FORMAT(1HO,'TAKE HALF STEP')
  806 FORMAT(1HO,'GRADIENT SCALAR PRODUCT NEGATIVE')
  807 FORMAT(1HO,'STEM = ',F5.2)
  808 FORMAT(1HO,'MODEL PI = ',E20.8)
  809 FORMAT(1HO,'STEP SIZE DOUBLED')
  810 FORMAT(1HO,'WALK ALONG VALLEY')
 1000 FORMAT(1HO,'DAP')
 1001 FORMAT(1HO,'DXP')
 1002 FORMAT(1HO,'DAS')
 1003 FORMAT(1HO,'DXS')
 1004 FORMAT(1HO,'DAPS')
```

```
 1005 FORMAT(1HO,'DXPS')
C
C      READ INPUT DATA
C
      READ(5,3) N,M,K,STEP,ITMAX,H,DIL,EPS,KK
      READ(5,24) (PAR(I),  I=1,K)
      READ(5,24) (ALPHA(I),I=1,N)
      IF(KK.EQ.0) GO TO 241
      KKS=KK*KK
      READ(5,24) (R(I),I=1,KKS)
      READ(5,24) (SI(I),I=1,KK)
  241 CONTINUE
C
C     INITIALIZE VARIABLES.
C
      PIOLD=0.
      IT=0
      ISP=0
      DO 27 I=1,K
      PAROLD(I)=0.
   27 CONTINUE
      NPO=N+1
      NMO=N-1
      NN=N*N
      NNN=N*NN
      NNPO=NN+1
      NMON=NMO*NN
      NNMO=N*NMO
      NKK=N*KK
      KKK=K*KK
      NK=N*K
      NKKK=NKK*K
      NMM=N-M
      DPI=0.
C
C
```

```
C       COMPUTE THE WEIGHTING MATRIX OF THE MODEL PERFORMANCE INDEX
C
      DO 240 I=1,N
      NI=N*(I-1)
      DO 240 J=1,N
      NIJ=NI+J
      Q(NIJ)=ALPHA(I)*ALPHA(J)/(ALPHA(1)**2)
  240 CONTINUE
      WRITE(6,1)
C
C       WRITE INPUT DATA
C
      WRITE(6,103)
      WRITE(6,3) N,M,K,STEP,ITMAX,H,DIL,EPS,KK
      WRITE(6,109)
      WRITE(6,20)  (ALPHA(I), I=1,N)
      WRITE(6,106)
      WRITE(6,20)  (Q(I),I=1,NN)
      WRITE(6,107)
      WRITE(6,20)  (PAR(I),  I=1,K)
      IF(KK.EQ.0)  GO TO 242
      WRITE(6,104)
      WRITE(6,20)  (R(I),I=1,KKS)
      WRITE(6,105)
      WRITE(6,20)  (SI(I),I=1,KK)
  242 CONTINUE
C
C
C       START OF GRADIENT ALGORITHM
C
C
C
      MPI=0
      IND=0
      NSTEP=0
      IFLAG=0
```

```
      GSQ=1.
  490 CONTINUE
C
      DO 620 I=1,NN
      U(I)=0.
  620 CONTINUE
      WRITE(6,12) STEP
      WRITE(6,110)
      WRITE(6,20) (PAR(I), I=1,K)
C
C      COMPUTE SYSTEM COEFFICIENTS, INITIAL CONDITIONS AND THEIR DERIVATIVES
C      USING SYST
C
      CALL SYST(ACOF,BCOF,X0,PAR,SI)
      WRITE(6,995)
      WRITE(6,20) (ACOF(I),I=1,N)
      WRITE(6,994)
      WRITE(6,20) (X0(I), I=1,N)
      IF(IT.GT.0) GO TO 300
      WRITE(6,1000)
      WRITE(6,24) (DAP(I),I=1,NK)
      WRITE(6,1001)
      WRITE(6,24) (DXP(I),I=1,NK)
      WRITE(6,1002)
      WRITE(6,24) (DAS(I),I=1,NKK)
      WRITE(6,1003)
      WRITE(6,24) (DXS(I),I=1,NKK)
      WRITE(6,1004)
      WRITE(6,24) (DAPS(I),I=1,NKKK)
      WRITE(6,1005)
      WRITE(6,24) (DXPS(I),I=1,NKKK)
  300 CONTINUE
C
C
C      COMPUTE THE OUTER PRODUCT OF X0 AND ITS TRANSPOSE
C
```

```
      DO 322 I=1,N
      NI=N*(I-1)
      DO 322 J=1,N
      NIJ=NI+J
      XXO(NIJ)=XO(I)*XO(J)
  322 CONTINUE
C
C       COMPUTE THE SOLUTIONS OF THE X AND P2 MATRICES USING STST
C
      CALL STST(ACOF,Q,XXO,P2,XX,N,0,0)
      IF(KK.EQ.0) GO TO 630
      CALL GMPRD(DAS,R,SR,N,KK,KK)
      CALL GMTRA(DAS,ST,N,KK)
      CALL GMPRD(SR,ST,W,N,KK,N)
      DO 602 I=1,NN
      W(I)=-W(I)
  602 CONTINUE
      CALL GMPRD(XX,W,XW,N,N,N)
      CALL GMPRD(DXS,R,TR,N,KK,KK)
      CALL GMTRA(DXS,TT,N,KK)
      CALL GMPRD(TR,TT,V,N,KK,N)
      CALL GMPRD(SR,TT,UV,N,KK,N)
C
C       COMPUTE THE SOLUTIONS OF THE Z(I) AND LAMBDA(I) MATRICES USING RTRT
C
      CALL RTRT(ACOF,XW,UV,P2,PZ,P,TZ,D,PV,PXO,XO)
      DO 601 I=1,NMO
      NI=N*I
      NMI=NNMO+I
      U(NI)=-TZ(I)
      U(NMI)=-TZ(I)
  601 CONTINUE
      U(NN)=-2.*TZ(N)
      DO 604 I=1,NN
      U(I)=U(I)+V(I)
  604 CONTINUE
```

```
      CALL GMPRD(W,P,WP,N,N,N)
C
C      COMPUTE THE SOLUTIONS OF THE DX AND P1 MATRICES USING STST
C
      CALL STST(ACOF,WP,U,P1,DX,N,0,1)
  630 CONTINUE
C
C      COMPUTE THE VALUE OF THE PERFORMANCE INDEX
C      TQX= TRACE( Q*XX ) AND TQDX= TRACE ( Q*DX )
C
      TQX=0.
      TQDX=0.
      DO 606 I=1,N
      NI=N*(I-1)
      DO 606 J=1,N
      NIJ=NI+J
      NJ=N*(J-1)
      NJI=NJ+I
      TQX=TQX+Q(NJI)*XX(NIJ)
      IF(KK.EQ.0) GO TO 606
      TQDX=TQDX+Q(NJI)*DX(NIJ)
  606 CONTINUE
      PI=TQX+EPS*TQDX
      DPI=PI-PIOLD
      WRITE(6,993)
      WRITE(6,20) PI,DPI,TQX,TQDX
      IF((TQX.GT.0.).AND.(TQDX.GT.0.)) GO TO 607
      WRITE(6,9)
      GO TO 80
  607 CONTINUE
      IF(ISP.EQ.1) GO TO 420
C
C      CONTROL OF THE GRADIENT STEP SIZE
C
      IF(IT.EQ.0) GO TO 460
      IF(IFLAG.NE.2) GO TO 400
```

```
      IF(NSTEP.NE.0) GO TO 400
      IF(IND.NE.0) GO TO 400
      GO TO 420
  400 CONTINUE
      IF(DPI.GE.0.) GO TO 421
C
C      TERMINATE THE PROCESS IF PREDICTED AND ACTUAL CHANGES IN PI ARE
C      SMALLER THAN H*PI
C
      IF(ABS(DPI).GE.(H*PI)) GO TO 440
      IF(ABS(EPI).GE.(H*PI)) GO TO 440
      IF(IFLAG.NE.1) GO TO 450
      IFLAG=2
      GO TO 460
  440 CONTINUE
      IF(IND.GT.0) GO TO 430
      IFLAG=0
C
C      COMPUTE THE DIFFERENCE BETWEEN ACTUAL AND PREDICTED CHANGE IN PI.
C      DOUBLE STEP SIZE IF DIFF IS LESS THAN DIL, THE SPECIFIED VALUE
C
      DIFF=ABS((DPI-EPI)/EPI)
      IF(DIFF.GT.DIL) GO TO 460
      STEP=2.*STEP
      WRITE(6,809)
      GO TO 460
  430 CONTINUE
      IND=0
      IFLAG=1
      GO TO 460
  421 CONTINUE
      IF(NSTEP.EQ.0) GO TO 420
      IFLAG=1
      NSTEP=0
      IND=0
      GO TO 460
```

```
  420 CONTINUE
C
C      COMPUTE PARABOLIC STEP SIZE
C
      ISP=0
      GD=0.
      DO 422 I=1,K
      GD=GD+GPI(I)*GSTEP(I)
  422 CONTINUE
      DGD=-2.*(GD-(PI-PIOLD))
      IF(DGD.LE.0.) GO TO 423
      STEM=-GD/DGD
      IF(STEM.GT.1.5) STEM=1.5
      IF(STEM.LT..1) STEM=0.1
C
C      STEM MULTIPLIES THE PREVIOUS STEP VALUE
C
      STEP=STEP*STEM
      WRITE(6,802)
      WRITE(6,807) STEM
      GO TO 424
  423 CONTINUE
C
C      CUT STEP SIZE IN HALF IF PARABOLIC STEP FAILED
C
      STEP=STEP/2.
      WRITE(6,805)
  424 CONTINUE
      IND=IND+1
C
C      TERMINATE IF PARABOLIC STEP FAILS THREE CONSECUTIVE TIMES
C
      IF(IND.LE.3) GO TO 470
      WRITE(6,803)
      GO TO 80
  460 CONTINUE
```

```
C
C     TERMINATE IF THE NUMBER OF ITERATIONS EXCEEDS ITMAX.
C
      IF(ITMAX-IT) 72,72,71
C
   71 WRITE(6,11) IT
C
C     COMPUTE THE GRADIENT
C
C      NOTE THAT CROSS DERIVATIVE MATRICES ARE STORED AS SINGLE SUB-
C      SCRIPTED VECTORS
C
      CALL GMPRD(P2,DXP,P2D,N,N,K)
      IF(KK.EQ.0) GO TO 631
C
C      CONTRIBUTION OF LAMBDA(I)*XO*VT(I-1)
C
      CALL GMPRD(PV,DXP,FF,1,N,K)
      NCOUNT=1
  702 CONTINUE
      INC=N*(NCOUNT-1)
      DO 700 J=1,K
      NKJ=NKK*(J-1)
      DO 701 I=1,N
      DVP(I)=0.
      DO 701 L=1,KK
      NL=N*(L-1)
      NLI=NL+I
      NKL=NKJ+NL
      DVP(I)=DVP(I)+DXPS(NCOUNT+NKL)*SR(NLI)+TR(NL+NCOUNT)*DAPS(NKL+I)
  701 CONTINUE
      DO 703 I=1,N
      FF(J)=FF(J)+PXO(INC+I)*DVP(I)
  703 CONTINUE
  700 CONTINUE
      IF(NCOUNT.GT.1) GO TO 704
```

```
      NCOUNT=NMM
  704 CONTINUE
      IF(NCOUNT.GE.N) GO TO 705
      NCOUNT=NCOUNT+1
      GO TO 702
  705 CONTINUE
C
C      CONTRIBUTION OF LAMBDA(N)*X*W, P1*XO AND P2*DXO
C
      CALL GMPRD(P,XX,PX,N,N,N)
      DO 600 I=1,N
      NI=N*(I-1)
      DO 600 J=1,N
      NIJ=NI+J
      JIN=N*(J-1)+I
      PXX(NIJ)=(PX(NIJ)+PX(JIN))/2.
      P1S(NIJ)=(P1(NIJ)+P1(JIN))/2.
  600 CONTINUE
      CALL GMPRD(P1S,DXP,P1D,N,N,K)
      DO 640 I=1,NK
      P2D(I)=P2D(I)+EPS*P1D(I)
  640 CONTINUE
      CALL GMPRD(PXX,SR,PX,N,N,KK)
      CALL GMPRD(P2,TR,PT,N,N,KK)
      DO 609 I=1,K
      EE(I)=0.
      NKI=NKK*(I-1)
      DO 610 J=1,NKK
      NKIJ=NKI+J
      EE(I)=EE(I)+DAPS(NKIJ)*PX(J)-DXPS(NKIJ)*PT(J)
  610 CONTINUE
      EE(I)=2.*EE(I)-FF(I)
  609 CONTINUE
C
C     CONTRIBUTION OF (P1*A+AT*P1)*X AND (P2*A+AT*P2)*DX
C
```

```
  631 CONTINUE
      DO 611 I=1,N
      DD(I)=0.
      NI=N*(I-1)
      DO 612 J=1,N
      DD(I)=DD(I)+P2(J*N)*XX(NI+J)
  612 CONTINUE
      IF(KK.EQ.0) GO TO 615
      DO 616 J=1,N
      JN=J*N
      NIJ=NI+J
      DD(I)=DD(I)+EPS*(P1S(JN)*XX(NIJ)+P2(JN)*DX(NIJ))
  616 CONTINUE
      DD(I)=2.*DD(I)+(PZ(I)+D(I))*EPS
      GO TO 611
  615 CONTINUE
      DD(I)=2.*DD(I)
  611 CONTINUE
C
C     ADD ALL TERMS TO FORM GRADIENT
C
      DO 613 I=1,K
      NI=N*(I-1)
      GPI(I)=0.
      DO 614 J=1,N
      NIJ=NI+J
      GPI(I)=GPI(I)-DD(J)*DAP(NIJ)+2.*XO(J)*P2D(NIJ)
  614 CONTINUE
  613 CONTINUE
      IF(KK.EQ.0) GO TO 618
      DO 617 I=1,K
      GPI(I)=GPI(I)-EPS*EE(I)
  617 CONTINUE
  618 CONTINUE
C
C      GRADIENT WITH RESPECT TO PERCENTAGE CHANGES IN PAR
```

```
C
      DO 619 I=1,K
      GPI(I)=PAR(I)*GPI(I)
  619 CONTINUE
      WRITE(6,125)
      WRITE(6,20) (GPI(I), I=1,K)
      IF(IT.EQ.0) GO TO 544
C
C      COMPUTE SCALAR PRODUCT OF GRADIENTS
C
      GSQ=0.
      GTG=0.
      DO 543 I=1,K
      GSQ=GSQ+GPI(I)*GPI(I)
      GTG=GTG+GPI(I)*GPIOLD(I)
  543 CONTINUE
      IF(GTG.GE.0.) GO TO 544
      ISP=1
      WRITE(6,806)
  544 CONTINUE
C
C     DIRECTION COSINE OF GRADIENT
C
      GPIEN=SQRT(GSQ)
      DO 542 I=1,K
      GPIOLD(I)=GPI(I)/GPIEN
  542 CONTINUE
      WRITE(6,804)
      WRITE(6,20) (GPIOLD(I),I=1,K)
C
C      SAVE OLD PARAMETER VALUES AND TAKE GRADIENT STEP
C
      IT=IT+1
      PIOLD=PI
      DO 461 I=1,K
      PAROLD(I)=PAR(I)
```

```
461 CONTINUE
470 CONTINUE
    DO 471 I=1,K
    GSTEP(I)=-STEP*GPIOLD(I)
    PAR(I)=PAROLD(I)+GSTEP(I)
471 CONTINUE
    EPI=-GPIEN*STEP
    GO TO 490
450 CONTINUE
    WRITE(6,7)
    GO TO 80
 72 WRITE(6,5)
 80 CONTINUE
    WRITE(6,11)  IT
    WRITE(6,12)  STEP
    WRITE(6,8)
    WRITE(6,20)  (PAR(I), I=1,K)
    RETURN
    END
```

```
C ******************************************************************************
C
C     THORGEIR PALSSON, MIT DEPT. OF AERONAUTICS AND ASTRONAUTICS, MAY 1970
C
C ******************************************************************************
C
C     SUBROUTINE STST
C
C     PURPOSE
C
C     COMPUTES THE SOLUTIONS OF THE EQUATIONS A*XX+XX*AT = Q AND AT*P+P*A = C
C     SIMULTANEOUSLY FOR A IN THE PHASE VARIABLE FORM
C
C     USAGE
C
C     CALL STST(ACOF,C,Q,P,XX,N,MST,INV)
C
C     DESCRIPTION OF INPUT PARAMETERS
C
C     ACOF = NEGATIVE OF THE LAST ROW OF A ( N COEFFICIENTS OF CHARACTERISTIC
C            EQUATION)
C     C AND Q = CONSTANT N*N MATRICES IN GENERAL OR SYMMETRIC STORAGE MODE
C     N = DIMENSION OF SYSTEM MATRICES
C     MST = 0   GENERAL STORAGE MODE
C     MST = 1   SYMMETRIC STORAGE MODE
C     INV = 0   INVERSE OF E MATRIX COMPUTED
C     INV = 1   INVERSE OF E MATRIX PROVIDED
C
C     DESCRIPTION OF OUTPUT PARAMETERS
C
C     P AND XX = N*N SOLUTION MATRICES IN GENERAL OR SYMMETRIC STORAGE MODE
C
C     REMARKS
C
C     NOTE THAT THE INVERSE OF THE E MATRIX IS IN COMMON WITH THE CALLING
C     PROGRAM, WHICH MUST HAVE A CORRESPONDING COMMON STATEMENT
```

```
C
C     THE PROGRAM PRINTS OUT THE  MAXIMUM RELATIVE ERROR IN THE SOLUTION OF THE
C     EQUATIONS FOR EACH ITERATION AS WELL AS THE NUMBER OF ITERATIONS PERFORMED
C     THE DETERMINANT OF THE E MATRIX IS ALSO PRINTED UPON INVERSION OF THAT
C     MATRIX.
C
C     SUBROUTINES REQUIRED
C
C     SNORM -   COMPUTES SUP-NORM OF AN N*N MATRIX
C     MINV -   MATRIX INVERSION ROUTINE FROM SSP LIBRARY
C
C ******************************************************************************
      SUBROUTINE STST(ACOF,C,Q,P,XX,N,MST,INV)
      DIMENSION AN(400),BN(400),AM(20),PNQ(20),PFD(20),DD(20)
      DIMENSION QQ(400),CC(400),LL(20),MM(20)
      DIMENSION C(1),Q(1),ACOF(1),P(1),XX(1)
      DIMENSION QA(400),CA(400),ACC(400),AQQ(400),AAN(400),ABN(400)
      DIMENSION TCA(400),TQA(400)
      DOUBLE PRECISION AAN,ABN,TCA,TQA,PFD,PNQ,AN,BN
      DOUBLE PRECISION DD,AM,E,QA,CA,DET
  153 FORMAT(1H0,5E16.8)
      COMMON/EINV/E(400)
C
C     COMPUTE CONSTANTS AND SET INITIAL CONDITIONS
C
      NN=N*N
      NPO=N+1
      NMO=N-1
      NMT=N-2
      NNMO=N*(N-1)
      NNPO=NNMO+1
      DET=0.
      IF(MST.NE.0) GO TO 155
      DO 156 I=1,NN
      CC(I)=-C(I)
      QQ(I)=-Q(I)
```

```
  156 CONTINUE
      GO TO 152
  155 CONTINUE
C
C     CHANGE STORAGE MODE OF FORCING MATRICES FROM SYMMETRIC TO GENERAL
C
      LOC=0
      DO 151 JCOUNT=1,N
      JCPO=JCOUNT+1
      JCMO=JCOUNT-1
      NJC=N*JCMO
      LOC=LOC+JCMO
      DO 150 I=1,JCOUNT
      CC(NJC+I)=-C(LOC+I)
      QQ(NJC+I)=-Q(LOC+I)
  150 CONTINUE
      LLOC=LOC+JCOUNT
      INC=JCOUNT
      IF(JCPO.GT.N) GO TO 152
      DO 151 I=JCPO,N
      LLOC=LLOC+INC
      CC(NJC+I)=-C(LLOC)
      QQ(NJC+I)=-Q(LLOC)
      INC=INC+1
  151 CONTINUE
  152 CONTINUE
C
C     STORE CC AND QQ IN ACC AND AQQ
C
      DO 410 I=1,NN
      ACC(I)=CC(I)
      AQQ(I)=QQ(I)
      AAN(I)=0.
      ABN(I)=0.
      TCA(I)=0.
      TQA(I)=0.
```

```
  410 CONTINUE
C
C     COMPUTE RIGHT HAND SIDE OF EQUATIONS FOR FIRST COLUMN OF SOLUTION
C     MATRICES
C
      NREF=0
  140 CONTINUE
      ICOUNT=1
      DO 100 I=1,NN
      AN(I)=0.
  100 CONTINUE
      DO 101 I=1,NN,NPO
      AN(I)=1.
  101 CONTINUE
      DO 102 I=1,N
      AM(I)=0.
      PFD(I)=0.
  102 CONTINUE
  130 CONTINUE
      NNI=N*(N-ICOUNT)
      DO 120 I=1,N
      PNQ(I)=0.
      DD(I)=PFD(I)+CC(NNI+I)
  120 CONTINUE
      DO 104 I=1,N
      DO 104 J=1,N
      PNQ(I)=PNQ(I)+AN(I+N*(J-1))*QQ(NNI+J)
  104 CONTINUE
      DO 121 I=1,N
      AM(I)=AM(I)+PNQ(I)
  121 CONTINUE
      IF(ICOUNT.EQ.N) GO TO 170
      PFD(1)=ACOF(1)*DD(N)
      DO 160 I=2,N
      PFD(I)=-DD(I-1)+ACOF(I)*DD(N)
  160 CONTINUE
```

```
  170 CONTINUE
C
C     COMPUTE THE  E MATRIX AND FIND THE INVERSE
C
      DO 106 I=1,NMO
      NPI=NNMO+I
      DO 106 J=I,NPI,N
      BN(J)=-AN(J+1)
  106 CONTINUE
      DO 107 I=N,NN,N
      BN(I)=0.
      DO 107 J=1,N
      INJ=I+J-N
      BN(I)=BN(I)+ACOF(J)*AN(INJ)
  107 CONTINUE
      NPIC=NPO-ICOUNT
      DO 110 I=1,NN,NPO
      BN(I)=BN(I)+ACOF(NPIC)
  110 CONTINUE
      IF(ICOUNT.EQ.N) GO TO 111
      DO 109 I=1,NN
      AN(I)=BN(I)
  109 CONTINUE
      ICOUNT=ICOUNT+1
      IF(ICOUNT.LE.N) GO TO 130
  111 CONTINUE
      IF((NREF.GT.0).OR.(INV.GT.0)) GO TO 500
      DO 112 I=1,NN
      E(I)=BN(I)
  112 CONTINUE
      CALL MINV(E,N,DET,LL,MM)
  400 FORMAT(1H0,'DETERMINANT OF E =',E16.5)
      WRITE(6,400) DET
C
C     COMPUTE THE FIRST AND LAST COLUMNS OF THE SOLUTION MATRICES AND
C     STORE THEM IN AN AND BN RESPECTIVELY
```

```
C
  500 CONTINUE
      DO 200 I=1,N
      AN(I)=0.
      DO 200 J=1,N
      AN(I)=AN(I)-E(N*(J-1)+I)*AM(J)
  200 CONTINUE
      KK=0
      DO 205 I=NNPO,NN
      BN(I)=0.
      DO 204 J=1,N
      BN(I)=BN(I)-E(J+KK)*DD(J)
  204 CONTINUE
      KK=KK+N
  205 CONTINUE
C
C     COMPUTE THE REMAINING COLUMNS OF THE SOLUTION MATRICES
C
      DO 207 I=N,NNMO,N
      IPN=I+N
      IMN=I-N
      DO 203 J=1,NMO
      IPJ=I+J
      IPJN=IPJ-NMO
      AN(IPJ)=-AN(IPJN)+QQ(IPJN-1)
  203 CONTINUE
      AN(IPN)=0.
      DO 202 L=1,N
      AN(IPN)=AN(IPN)+ACOF(L)*AN(IMN+L)
  202 CONTINUE
      AN(IPN)=AN(IPN)+QQ(I)
  207 CONTINUE
      DO 206 I=N,NNMO,N
      NOMI=NNMO-I
      NNMI=NN-I
      NIPN=NNMI+N
```

```
      NIDN=NIPN/N
      BN(NOMI+1)=ACOF(NIDN)*BN(NNMO+1)+ACOF(1)*BN(NIPN)+CC(NNMI+1)
      DO 206 J=2,N
      BN(NOMI+J)=ACOF(NIDN)*BN(NNMO+J)+ACOF(J)*BN(NIPN)-BN(NNMI+J-1)+CC
     1(NNMI+J)
  206 CONTINUE
C
C     SUBSTITUTE SOLUTIONS INTO EQUATIONS AND COMPUTE THE RHS
C
      DO 401 I=1,NN
      QA(I)=0.
      CA(I)=0.
  401 CONTINUE
      DO 402 I=1,N
      NIMO=N*(I-1)
      CA(1+NIMO)=-ACOF(1)*BN(I*N)
      DO 402 J=2,N
      CA(J+NIMO)=-ACOF(J)*BN(I*N)+BN(J-1+NIMO)
  402 CONTINUE
      DO 403 J=1,N
      CA(J)=CA(J)-ACOF(1)*BN(NNMO+J)
      DO 403 I=1,NMO
      CA(J+I*N)=CA(J+I*N)-ACOF(I+1)*BN(NNMO+J)+BN(N*(I-1)+J)
  403 CONTINUE
      DO 404 I=1,N
      NIMO=N*(I-1)
      DO 411 J=1,NMO
      QA(NIMO+J)=AN(NIMO+J+1)
  411 CONTINUE
      DO 404 K=1,N
      QA(N*I)=QA(N*I)-ACOF(K)*AN(K+NIMO)
  404 CONTINUE
      DO 405 J=1,N
      DO 406 I=1,NMO
      NIMO=N*(I-1)
      QA(NIMO+J)=QA(NIMO+J)+AN(J+I*N)
```

```
  406 CONTINUE
      DO 405 K=1,N
      QA(N*NMO+J)=QA(N*NMO+J)-ACOF(K)*AN(J+N*(K-1))
  405 CONTINUE
C
C     ADD SOLUTIONS TO ANY PREVIOUS SOLUTION STORED IN AAN AND ABN
C
      DO 409 I=1,NN
      AAN(I)=AAN(I)+AN(I)
      ABN(I)=ABN(I)+BN(I)
  409 CONTINUE
C
C     ADD THE COMPUTED RHS TO ANY PREVIOUSLY COMPUTED RHS MATRICES STORED IN
C     TCA AND TQA. THEN FIND THE DIFFERENCE BETWEEN THE COMPUTED RHS AND THE
C     ACTUAL VALUES
C
      DO 407 I=1,NN
      TCA(I)=TCA(I)+CA(I)
      TQA(I)=TQA(I)+QA(I)
      CA(I)=ACC(I)-TCA(I)
      QA(I)=AQQ(I)-TQA(I)
      AN(I)=CA(I)
      BN(I)=QA(I)
C
C     FIND THE RELATIVE MAGNITUDE OF THE ERROR IN THE COMPUTED RHS
C
      IF(ABS(ACC(I)).GT.(.1E-10)) GO TO 430
      AN(I)=0.
      GO TO 432
  430 CONTINUE
      AN(I)=AN(I)/ACC(I)
  432 CONTINUE
      IF(ABS(AQQ(I)).GT.(.1E-10)) GO TO 431
      BN(I)=0.
      GO TO 433
  431 CONTINUE
```

```
      BN(I)=BN(I)/AQQ(I)
  433 CONTINUE
  407 CONTINUE
      CALL SNORM(AN,SCA,N,0)
      CALL SNORM(BN,SQA,N,0)
  157 FORMAT(1HO,'NORMALIZED ERRORS')
      WRITE(6,157)
      WRITE(6,153) SCA,SQA
      DO 440 I=1,NN
      CC(I)=CA(I)
      QQ(I)=QA(I)
  440 CONTINUE
C
C     IF ERROR IS GREATER THAN 1.0E-10 PERFORM AN ITERATION
C
      IF(SCA.GT.(1.0E-10)) GO TO 420
      IF(SQA.GT.(1.0E-10)) GO TO 420
      GO TO 408
  420 CONTINUE
      NREF=NREF+1
      IF(NREF.GT.10) GO TO 408
      GO TO 140
  408 CONTINUE
  154 FORMAT(1HO,'NO OF ITERATIONS = ',I3)
      WRITE(6,154) NREF
      IF(MST.EQ.0) GO TO 305
C
C     PLACE SOLUTIONS IN XX AND P  IN A SYMMETRIC STORAGE MODE
C     USE AVERAGES OF OFF-DIAGONAL TERMS
C
      LKI=0
      NJI=0
      LK=0
      DO 300 J=1,N
      JMO=J-1
      LK=LK+JMO
```

```
      IF(JMO.EQ.0) GO TO 304
      DO 301 I=1,JMO
      NJI=N*JMO+I
      JIN=N*(I-1)+J
      LKI=LK+I
      XX(LKI)=(AAN(NJI)+AAN(JIN))/2.
      P(LKI)=(ABN(NJI)+ABN(JIN))/2.
  301 CONTINUE
  304 CONTINUE
      XX(LKI+1)=AAN(NJI+1)
      P(LKI+1)=ABN(NJI+1)
  300 CONTINUE
      GO TO 306
  305 CONTINUE
      DO 307 I=1,N
      NI=N*(I-1)
      DO 307 J=1,N
      NIJ=NI+J
      NJI=N*(J-1)+I
      XX(NIJ)=(AAN(NIJ)+AAN(NJI))/2.
      P(NIJ)=(ABN(NIJ)+ABN(NJI))/2.
  307 CONTINUE
  306 CONTINUE
      RETURN
      END
```

```
C ******************************************************************************
C
C      THORGEIR PALSSON, MIT DEPT. OF AERONAUTICS AND ASTRONAUTICS, 1971
C
C ******************************************************************************
C
C
C      SUBROUTINE RTRT
C
C      PURPOSE
C
C      COMPUTES THE SOLUTIONS OF THE MATRIX EQUATIONS FOR Z(I) AND LAMBDA(I)
C      USING THE METHODS OF CHAPTER 4 IN THESIS
C
C      INPUT PARAMETERS
C
C      ACOF = CLOSED LOOP DENOMINATOR COEFFICIENTS
C      XW = PRODUCT OF X AND W MATRICES
C      UV = CROSS CORRELATION OF DA AND DX
C      P2 = SOLUTION MATRIX FROM STST
C      X0 = SYSTEM INITIAL CONDITION VECTOR
C
C      OUTPUT PARAMETERS
C
C      PZ = LAST COLUMN VECTOR OFTHE MATRIX OBTAINED BY SUMMING ALL
C           PRODUCTS OF THE FORM Z(I)*LAMBDA(I+1)
C      PI = LAMBDA(N) MATRIX
C      TZ(I)= TRACE OF Z(I-1)
C      D(I) = TRACE OF THE PRODUCT LAMBDA(N)*Z(I-1)
C      PV = VECTOR SUM OF ALL THE PRODUCTS LAMBDA(I)*V(I-1)
C      PX0 = MATRIX WHOSE COLUMNS ARE PRODUCTS OF THE FORM LAMBDA(I)*X0
C
C      REMARKS
C
C      NOTE THAT PI NOTATION IS USED HERE INSTEAD OF THE LAMBDA NOTATION IN
C      THESIS
```

```
C
C      SUBROUTINES REQUIRED
C
C      SNORM
C      MPROD ( DOUBLE PRECISION VERSION OF GMPRD )
C
C *************************************************************************
C
C
      SUBROUTINE RTRT(ACOF,XW,UV,P2,PZ,PI,TZ,D,PV,PXO,XO)
      DIMENSION AN(400),BN(400),DN(400),Z(4000),ZO(400),ACOF(20)
      DIMENSION P2(400),XW(400),TZ(20),PZ(20),D(20),PI(400),AA(400)
      DIMENSION UV(1),PV(1),PXO(1),XO(1),DD(20)
      DIMENSION API(400),AP(400)
      DOUBLE PRECISION AN,BN,DN,AP,API,E,Z,ZO
      COMMON/CONST/N,M,K,KK,KKK,NN,NNN,NPO,NMO,NNPO,NNMO,NMON,NKK
      COMMON/EINV/E(400)
   10 FORMAT(1HO,'NORMALIZED ERROR IN P =',E16.8)
  117 FORMAT(1HO,'NORMALIZED ERROR IN Z =',E16.8)
C
C      INITIALIZATION
C
      IREF=0
      NMM=N-M
      DO 106 I=1,NNN
      Z(I)=0.
  106 CONTINUE
      DO 115 I=1,N
      PZ(I)=0.
      TZ(I)=0.
      PV(I)=0.
  115 CONTINUE
      IC=1
      DO 160 I=1,NN
      AN(I)=0.
      PI(I)=0.
```

```
      AP(I)=0.
  160 CONTINUE
C
C     COMPUTE D(O)*XO
C
      DO 161 I=1,N
      DD(I)=XO(I)
      D(I)=XO(I)
  161 CONTINUE
      IF(M.EQ.0) GO TO 163
  164 CONTINUE
C
C     COMPUTE D(O)*XO*V(N-I)
C
      NIC=N*(N-IC)
      DO 162 I=1,N
      DO 162 J=1,N
      NJI=N*(J-1)
      AN(I+NJI)=AN(I+NJI)+DD(I)*UV(J+NIC)
  162 CONTINUE
      IF(IC.EQ.N) GO TO 168
  163 CONTINUE
      IC=IC+1
C
C     COMPUTE D(I)*XO
C
      NICO=N-IC+2
      DO 165 I=1,NMO
      DD(I)=-D(I+1)+ACOF(NICO)*XO(I)
  165 CONTINUE
      DD(N)=ACOF(NICO)*XO(N)
      DO 166 I=1,N
      DD(N)=DD(N)+ACOF(I)*D(I)
      D(I)=DD(I)
  166 CONTINUE
      IF((IC.LE.M).OR.(IC.EQ.N)) GO TO 164
```

```
      GO TO 163
  168 CONTINUE
      DO 167 I=1,NN
      BN(I)=XW(I)+AN(I)
      AA(I)=UV(I)
  167 CONTINUE
C
C     COMPUTE Z(0)
C
      CALL MPROD(E,BN,ZO,N,N,N)
C
C     STORE Z(0)
C
  107 CONTINUE
      DO 110 I=1,NN
      Z(I)=ZO(I)+Z(I)
      AN(I)=ZO(I)
  110 CONTINUE
C
C     COMPUTE THE REMAINING Z MATRICES
C
      DO 111 IK=1,NMO
      NIK=NN*IK
      NKI=N*(IK-1)
      DO 112 I=1,NMO
      DO 112 J=1,N
      IJ=I+N*(J-1)
      BN(IJ)=-AN(IJ+1)-XO(I)*AA(NKI+J)
  112 CONTINUE
      DO 113 IL=1,N
      I=N*IL
      BN(I)=-XO(N)*AA(NKI+IL)
      IMN=I-N
      DO 113 J=1,N
      INJ=IMN+J
      BN(I)=BN(I)+ACOF(J)*AN(INJ)
```

```
  113 CONTINUE
C
C     STORE Z(I)
C
      DO 111 I=1,NN
      Z(NIK+I)=BN(I)+Z(NIK+I)
      AN(I)=BN(I)
  111 CONTINUE
C
C     SUBSTITUTE SOLUTIONS INTO LAST EQUATION, STORING
C     THE COMPUTED R.H.S. IN ZO
C
      DO 120 I=1,NMO
      NPI=NNMO+I
      DO 120 J=I,NPI,N
      ZO(J)=Z(NMON+J+1)
  120 CONTINUE
      DO 121 I=N,NN,N
      ZO(I)=0.
      IMN=I-N
      DO 121 J=1,N
      INJ=IMN+J
      ZO(I)=ZO(I)-ACOF(J)*Z(NMON+INJ)
  121 CONTINUE
      DO 122 I=1,N
      NI=NN*(I-1)
      DO 122 J=1,NN
      ZO(J)=ZO(J)-ACOF(I)*Z(NI+J)
  122 CONTINUE
      DO 124 I=1,N
      DO 124 J=1,N
      NJ=N*(J-1)
      ZO(I+NJ)=ZO(I+NJ)+XO(I)*UV(NNMO+J)
  124 CONTINUE
C
C     FIND MAXIMUM ERROR IN COMPUTED VALUE OF XW
```

```
C
      DO 123 I=1,NN
      BN(I)=ZO(I)+XW(I)
      AN(I)=0.
      IF(ABS(XW(I)).GT.(.1E-10)) AN(I)=BN(I)/XW(I)
  123 CONTINUE
      CALL SNORM(AN,SAN,N,0)
      WRITE(6,117) SAN
      IF(SAN.LT.(1.0E-10)) GO TO 130
      IREF=IREF+1
      IF(IREF.GT.10) GO TO 130
      DO 128 I=1,NN
      AA(I)=0.
  128 CONTINUE
C
C     COMPUTE CORRECTION FOR Z(0)
C
      CALL MPROD(E,BN,ZO,N,N,N)
      GO TO 107
  130 CONTINUE
C
C     COMPUTE THE PI MATRICES AND STORE THE SUM OF THE LAST COLUMNS
C     OF THE PRODUCTS Z*PI IN PZ
C
      DO 100 I=1,NN
      AN(I)=0.
      DN(I)=0.
  100 CONTINUE
      DO 101 I=1,NN,NPO
      AN(I)=1.
      DN(I)=P2(NNMO+1)
  101 CONTINUE
      DO 105 IK=2,N
      DO 102 I=1,NMO
      NPI=NNMO+I
      DO 102 J=I,NPI,N
```

```
      BN(J)=-AN(J+1)
  102 CONTINUE
      DO 103 I=N,NN,N
      BN(I)=0.
      IMN=I-N
      DO 103 J=1,N
      INJ=IMN+J
      BN(I)=BN(I)+ACOF(J)*AN(INJ)
  103 CONTINUE
      JNMO=NNMO+IK
      DO 104 I=1,NN
      DN(I)=DN(I)+P2(JNMO)*BN(I)
      AN(I)=BN(I)
  104 CONTINUE
  105 CONTINUE
      DO 116 I=1,NN
      DN(I)=-2.*DN(I)
  116 CONTINUE
C
C     COMPUTE PI(N)
C
      IREF=0
      CALL MPROD(DN,E,API,N,N,N)
      DO 125 I=1,NN
      PXO(I)=0.
  125 CONTINUE
C
C     COMPUTE LAST COLUMN OF Z(N-1)*PI(N)
C
  145 CONTINUE
      DO 126 I=1,N
      DO 126 J=1,N
      IJN=I+N*(J-1)
      PZ(I)=PZ(I)+Z(NMON+IJN)*API(NNMO+J)
  126 CONTINUE
C
```

```
C       COMPUTE PI(N)*XO AND VT(N-1)*PI(N)
C
        IF(M.EQ.0) GO TO 201
        DO 200 I=1,N
        INMO=I+NNMO
        NIO=N*(I-1)
        DO 200 J=1,N
        PXO(INMO)=PXO(INMO)+API(I+N*(J-1))*XO(J)
        PV(I)=PV(I)+UV(NNMO+J)*API(J+NIO)
  200   CONTINUE
  201   CONTINUE
C
C       COMPUTE PI(I)
C
        DO 131 I=1,NN
        AN(I)=API(I)
  131   CONTINUE
        DO 132 IK=2,N
        NIK=N-IK+2
        NNK=NN-IK+2
        NKN=NN*(N-IK)
        DO 133 J=1,N
        BN(J)=ACOF(NIK)*API(J)+ACOF(1)*AN(NNMO+J)
  133   CONTINUE
        DO 134 I=2,N
        NI=N*(I-1)
        DO 134 J=1,N
        NIJ=NI+J
        BN(NIJ)=ACOF(NIK)*API(NIJ)+ACOF(I)*AN(NNMO+J)-AN(NIJ-N)
  134   CONTINUE
        IF(IREF.GT.0) GO TO 144
        DO 135 I=1,NN,NPO
        BN(I)=BN(I)+2.*P2(NNK)
  135   CONTINUE
  144   CONTINUE
C
```

```
C      COMPUTE LAST COLUMN OF Z(I)*PI(I+1)
C
       DO 136 I=1,N
       DO 136 J=1,N
       IJN=I+N*(J-1)
       PZ(I)=PZ(I)+Z(NKN+IJN)*BN(NNMO+J)
  136  CONTINUE
C
C      COMPUTE PI(I)*XO AND VT(I-1)*PI(I)
C
       IF((IK.GT.M).AND.(IK.LT.N)) GO TO 205
       NMIK=N*(N-IK)
       DO 202 I=1,N
       NIO=N*(I-1)
       DO 202 J=1,N
       PXO(NMIK+I)=PXO(NMIK+I)+BN(I+N*(J-1))*XO(J)
       PV(I)=PV(I)+UV(NMIK+J)*BN(J+NIO)
  202  CONTINUE
  205  CONTINUE
       DO 139 I=1,NN
       AN(I)=BN(I)
  139  CONTINUE
  132  CONTINUE
       DO 146 I=1,NN
       PI(I)=PI(I)+API(I)
       AP(I)=AP(I)+BN(I)
  146  CONTINUE
C
C      TEST ACCURACY OF THE PI SOLUTION
C
       DO 140 I=1,N
       NPI=I+N*(I-1)
       D(I)=ACOF(1)*PI(NPI)+ACOF(I)*AP(NNMO+I)+2.*P2(NNMO+1)
  140  CONTINUE
       DO 141 I=2,N
       NPI=N*(I-2)
```

```
      D(I)=D(I)-AP(NPI+I)
  141 CONTINUE
      DMAX=0.
      TP=.5/P2(NNMO+1)
      DO 142 I=1,N
      D(I)=D(I)*TP
      AVD=ABS(D(I))
      IF(AVD.GT.DMAX) DMAX=AVD
  142 CONTINUE
      WRITE(6,10) DMAX
      IF(DMAX.LT.(1.0E-06)) GO TO 143
      IREF=IREF+1
      IF(IREF.GT.10) GO TO 143
      DO 170 I=1,N
      AN(I)=-ACOF(1)*(PI(I)+AP(NNMO+I))
  170 CONTINUE
      DO 171 I=1,NMO
      DO 171 J=1,N
      JIN=J+N*I
      AN(JIN)=-ACOF(1)*PI(JIN)-ACOF(I+1)*AP(NNMO+J)+AP(JIN-N)
  171 CONTINUE
      DO 172 I=1,NN,NPO
      AN(I)=AN(I)-2.*P2(NNMO+1)
  172 CONTINUE
C
C     COMPUTE CORRECTION FOR PI(N)
C
      CALL MPROD(AN,E,API,N,N,N)
      GO TO 145
  143 CONTINUE
C
C     COMPUTE TRACE(PI(N)*Z(I)) AND TRACE(Z(I)) AND STORE IN
C     D AND TZ RESPECTIVELY
C
      DO 150 I=1,N
      D(I)=0.
```

```
150 CONTINUE
    DO 151 I=1,N
    NNI=NN*(I-1)
    DO 151 J=1,N
    NJJ=J+N*(J-1)
    TZ(I)=TZ(I)+Z(NJJ+NNI)
    DO 151 IK=1,N
    NK=N*(IK-1)
    NIJ=IK+N*(J-1)
    D(I)=PI(J+NK)*Z(NNI+NIJ)+D(I)
151 CONTINUE
    RETURN
    END
```

```
C ***********************************************************************
C
C      THORGEIR PALSSON, MIT DEPT. OF AERONAUTICS AND ASTRONAUTICS, 1971
C
C ***********************************************************************
C
C
C      SUBROUTINE SYST
C
C      PURPOSE
C
C      COMPUTES THE COEFFICIENTS OF THE CLOSED LOOP TRANSFER FUNCTION,
C      THE SYSTEM INITIAL CONDITIONS AND THE REQUIRED DERIVATIVES OF THESE
C      QUANTITIES WITH RESPECT TO THE DESIGN PARAMETERS
C
C      INPUT PARAMETERS
C
C      PAR = VECTOR OF FREE DESIGN PARAMETERS
C      SI = VECTOR OF VARIABLE DESIGN PARAMETERS
C
C      OUTPUT PARAMETERS
C
C      ACOF = CLOSED LOOP DENOMINATOR COEFFICIENTS
C      BCOF = CLOSED LOOP NUMERATOR COEFFICIENTS
C      XO = SYSTEM INITIAL CONDITION VECTOR
C
C      SUBROUTINES REQUIRED
C
C      INCON
C      ROOTIN
C
C ***********************************************************************
C
C
      SUBROUTINE SYST(ACOF,BCOF,XO,PAR,SI)
      DIMENSION XO(20),PAR(10),SI(10),ACOF(20),BCOF(20)
```

```
      DIMENSION P(10),S(10),AA(80),XA(80),CS(4)
      COMMON/DERIV/DAP(200),DXP(200),DAS(200),DXS(200),DAPS(2000),DXPS(2
     1000)
      COMMON/CONST/N,M,K,KK,KKK,NN,NNN,NPO,NMO,NNPO,NNMO,NMON,NKK
C
C
C       INITIALIZE PARAMETERS
C
      N2=2*N
      N3=3*N
      DO 63 I=1,KK
      S(I)=SI(I)
   63 CONTINUE
      DO 64 I=1,K
      P(I)=PAR(I)
   64 CONTINUE
C
C
C       COMPUTE FIRST DERIVATIVES W.R.T. PAR
C
      DO 80 I=1,K
      ICOUNT=1
      P(I)=1.1*PAR(I)
   83 CONTINUE
      CALL ROOTIN(BCOF,ACOF,P,S)
      CALL INCON(BCOF,ACOF,XO)
      NIC=N*(ICOUNT-1)
      DO 81 J=1,N
      NIJ=NIC+J
      AA(NIJ)=ACOF(J)
      XA(NIJ)=XO(J)
   81 CONTINUE
      IF(ICOUNT.GT.1) GO TO 82
      ICOUNT=ICOUNT+1
      P(I)=.9*PAR(I)
      GO TO 83
```

```
   82 CONTINUE
      DPAR=.2*PAR(I)
      DO 84 J=1,N
      LOC=J+N*(I-1)
      DAP(LOC)=(AA(J)-AA(J+N))/DPAR
      DXP(LOC)=(XA(J)-XA(J+N))/DPAR
   84 CONTINUE
      P(I)=PAR(I)
   80 CONTINUE
C
C      COMPUTE CROSS DERIVATIVES OF ACOF AND XO WITH RESPECT TO PAR AND SI
C
      DO 75 I=1,KK
      IMO=I-1
      DO 74 J=1,K
      JMO=J-1
      S(I)=1.05*SI(I)
      ICOUNT=1
   71 CONTINUE
      P(J)=1.05*PAR(J)
   72 CONTINUE
      CALL ROOTIN(BCOF,ACOF,P,S)
      CALL INCON(BCOF,ACOF,XO)
C
      NIC=N*(ICOUNT-1)
      DO 76 IA=1,N
      NIA=NIC+IA
      AA(NIA)=ACOF(IA)
      XA(NIA)=XO(IA)
   76 CONTINUE
      IF(ICOUNT.NE.3) GO TO 78
      ICOUNT=ICOUNT+1
      GO TO 71
   78 CONTINUE
      IF(ICOUNT.NE.1) GO TO 77
      P(J)=.95*PAR(J)
```

```
      ICOUNT=ICOUNT+1
      GO TO 72
   77 CONTINUE
      IF(ICOUNT.NE.2) GO TO 73
      S(I)=.95*SI(I)
      ICOUNT=ICOUNT+1
      GO TO 72
   73 CONTINUE
      LOC=N*(KK*JMO+IMO)
      DEN=.01*PAR(J)*SI(I)
      DO 79 L=1,N
      KLOC=LOC+L
      DAPS(KLOC)=(AA(L)-AA(L+N)+AA(L+N2)-AA(L+N3))/DEN
      DXPS(KLOC)=(XA(L)-XA(L+N)+XA(L+N2)-XA(L+N3))/DEN
   79 CONTINUE
      P(J)=PAR(J)
   74 CONTINUE
      S(I)=SI(I)
   75 CONTINUE
C
C       COMPUTE FIRST DERIVATIVES W.R.T. SI
C
      DO 90 I=1,KK
      ICOUNT=1
      S(I)=1.1*SI(I)
   93 CONTINUE
      CALL ROOTIN(BCOF,ACOF,P,S)
      CALL INCON(BCOF,ACOF,XO)
      NIC=N*(ICOUNT-1)
      DO 91 J=1,N
      NIJ=NIC+J
      AA(NIJ)=ACOF(J)
      XA(NIJ)=XO(J)
   91 CONTINUE
      IF(ICOUNT.GT.1) GO TO 92
      ICOUNT=ICOUNT+1
```

```
      S(I)=.9*SI(I)
      GO TO 93
   92 CONTINUE
      DSI=.2*SI(I)
      DO 94 J=1,N
      LOC=J+N*(I-1)
      DAS(LOC)=(AA(J)-AA(J+N))/DSI
      DXS(LOC)=(XA(J)-XA(J+N))/DSI
   94 CONTINUE
      S(I)=SI(I)
   90 CONTINUE
C
C       COMPUTE NOMINAL COEFFICIENT VALUES
C
      CALL ROOTIN(BCOF,ACOF,PAR,SI)
      CALL INCON(BCOF,ACOF,XO)
   50 RETURN
      END
```

```
C ******************************************************************************
C
C     THORGEIR PALSSON, MIT DEPT. OF AERONAUTICS AND ASTRONAUTICS, 1971
C
C ******************************************************************************
C
C
C     SUBROUTINE ROOTIN
C
C     A MODIFIED VERSION OF A PROGRAM WRITTEN BY WILLIAM R. GRIFFIN,
C     MIT DEPT. OF AERONAUTICS AND ASTRONAUTICS
C
C     PURPOSE
C
C     FORMS CLOSED LOOP TRANSFER FUNCTIONS GIVEN OPEN LOOP ROOTS
C     AND STATIC SENSITIVITIES
C
C     INPUT PARAMETERS
C
C     PAR = VECTOR OF FREE DESIGN PARAMETERS
C     SI = VECTOR OF VARIABLE DESIGN PARAMETERS
C
C     OUTPUT PARAMETERS
C
C     AA = NUMERATOR COEFFICIENTS OF CLOSED LOOP TRANSFER FUNCTION
C     BB = DENOMINATOR COEFFICIENTS OF CLOSED LOOP TRANSFER FUNCTION
C
C     REMARKS
C
C     FOR EXPLANATION OF MODES REFER TO APPENDIX C OF THESIS
C
C     SUBROUTINES REQUIRED
C
C     SWEEP
C     ROOTS
C
```

```
C ***************************************************************************
C
C
      SUBROUTINE ROOTIN(AA,BB,PAR,SI)
      DIMENSION AA(20),BB(20)
      DIMENSION A(20),B(20),RPZ(20),RPP(20),CPZ(20),CPP(20),PAR(10)
      DIMENSION RPZB(20),RPPB(20),CPZB(20),CPPB(20),A2(20),B2(20)
      DIMENSION B3(20),A1(20),B1(20),AC(20),BC(20)
      DIMENSION RPPP(10),CPPP(10),RPZP(10),CPZP(10),BP(20),AP(20),SI(10)
      DIMENSION RPPC(10),CPPC(10),RPZC(10),CPZC(10)
      DIMENSION RPPD(10),CPPD(10),RPZD(10),CPZD(10),AD(20),BD(20)
      DIMENSION A3(20)
      COMMON / ROOT / RPP,CPP,RPZ,CPZ,RPPB,CPPB,RPZB,CPZB,RPPC,CPPC,RPZC
     1,CPZC,RPPD,CPPD,RPZD,CPZD,RPPP,CPPP,RPZP,CPZP,SA,SB,SC,SD,SP,NP,NZ
     2,NPB,NZB,NPC,NZC,NPD,NZD,NPP,NZP
      CALL ROOTS(PAR,SI,MODE)
      IF(MODE.EQ.2) GO TO 200
C
C     FORM NUMERATOR OF P
C
      CALL SWEEP(NZP,RPZP,CPZP,AP,NAP)
C
C     FORM DENOMINATOR OF P
C
      CALL SWEEP(NPP,RPPP,CPPP,BP,NBP)
      IF(MODE.NE.1) GO TO 300
      NA=NAP
      NB=NBP
      DO 302 I=1,NBP
      A(I)=AP(I)
      B(I)=BP(I)
  302 CONTINUE
      GO TO 320
  300 CONTINUE
C
C     MULTIPLY BY STATIC SENSITIVITY OF P
```

```
C
      DO 303 I=1,NAP
      AP(I)=SP*AP(I)
  303 CONTINUE
      IF(MODE.EQ.5) GO TO 400
  200 CONTINUE
C
C     FORM NUMERATOR OF A
C
      CALL SWEEP ( NZ, RPZ, CPZ, A1, NA1 )
      DO 301 I = 1, NA1
  301 A1(I)=SA*A1(I)
C
C     FORM DENOMINATOR OF B
C
      CALL SWEEP ( NPB, RPPB, CPPB, B1, NB1 )
C
C     FORM NUMERATOR OF AB LOOP
C
      CALL PMPY(AC,NAC,A1,NA1,B1,NB1)
C
C     FORM NUMERATOR OF B
C
      CALL SWEEP ( NZB, RPZB, CPZB, A2, NA2 )
      DO 210 I=1,NA2
      A2(I) = SB * A2(I)
  210 CONTINUE
C
C     FORM NUMERATOR OF A*B
C
      CALL PMPY ( B2, NB2 , A1, NA1, A2, NA2 )
C
C    FORM DENOMINATOR OF A
C
      CALL SWEEP ( NP, RPP, CPP, A1, NA1 )
C
```

```
C     FORM DENOMINATOR OF A*B
C
      CALL PMPY ( A2, NA2, A1, NA1, B1, NB1 )
C
C     FORM DENOMINATOR OF AB LOOP
C
      CALL PADD(BC,NBC,A2,NA2,B2,NB2)
      IF(MODE.GE.3) GO TO 321
  322 CONTINUE
      NA=NAC
      NB=NBC
      DO 330 I=1,NAC
  330 A(I)=AC(I)
      DO 331 I=1,NBC
  331 B(I)=BC(I)
      GO TO 320
  321 CONTINUE
C
C     MULTIPLY P*AB
C
      CALL PMPY(A,NA,AP,NAP,AC,NAC)
      CALL PMPY(B,NB,BP,NBP,BC,NBC)
      IF(MODE.NE.4) GO TO 320
C
C     FORM D TRANSFER FUNCTION AND MULTIPLY ABP*D
C
      CALL SWEEP(NPD,RPPD,CPPD,BD,NBD)
      CALL SWEEP(NZD,RPZD,CPZD,AD,NAD)
      DO 325 I=1,NAD
      AD(I)=SD*AD(I)
  325 CONTINUE
      CALL PMPY(AC,NAC,A,NA,AD,NAD)
      CALL PMPY(BC,NBC,B,NB,BD,NBD)
C
C     FORM TOTAL CLOSED LOOP TRANSFER FUNCTION
C
```

```
      CALL SWEEP(NPC,RPPC,CPPC,B1,NB1)
      CALL SWEEP(NZC,RPZC,CPZC,A1,NA1)
      DO 326 I=1,NA1
      A1(I)=SC*A1(I)
  326 CONTINUE
      CALL PMPY(A,NA,AC,NAC,B1,NB1)
      CALL PMPY(A2,NA2,BC,NBC,B1,NB1)
      CALL PMPY(B2,NB2,AC,NAC,A1,NA1)
      CALL PADD(B,NB,A2,NA2,B2,NB2)
      GO TO 320
  400 CONTINUE
C
C       FORM CLOSED LOOP TRANSFER FUNCTION FOR MODE=5
C
C       COMPUTE NUMERATORS OF A, B, AND C
C
C
      CALL SWEEP(NZ,RPZ,CPZ,A1,NA1)
      CALL SWEEP(NZB,RPZB,CPZB,A2,NA2)
      CALL SWEEP(NZC,RPZC,CPZC,A3,NA3)
      DO 401 I=1,NA1
      A1(I)=SA*A1(I)
      A2(I)=SB*A2(I)
      A3(I)=SC*A3(I)
  401 CONTINUE
C
C       COMPUTE DENOMINATORS OF A, B, AND C
C
      CALL SWEEP(NP,RPP,CPP,B1,NB1)
      CALL SWEEP(NPB,RPPB,CPPB,B2,NB2)
      CALL SWEEP(NPC,RPPC,CPPC,B3,NB3)
C
C
C       ADD A, B, AND C
C
      CALL PMPY(AC,NAC,A1,NA1,B2,NB2)
```

```
      CALL PMPY(AD,NAD,A2,NA2,B1,NB1)
      CALL PADD(A,NA,AC,NAC,AD,NAD)
      CALL PMPY(B,NB,B1,NB1,B2,NB2)
C
      CALL PMPY(AC,NAC,A,NA,B3,NB3)
      CALL PMPY(AD,NAD,A3,NA3,B,NB)
      CALL PADD(A1,NA1,AC,NAC,AD,NAD)
      CALL PMPY(B1,NB1,B,NB,B3,NB3)
C
C
C      MULTIPLY P*( A + B + C )
C
      CALL PMPY(AC,NAC,AP,NAP,A1,NA1)
      CALL PMPY(BC,NBC,BP,NBP,B1,NB1)
C
C
C      COMPUTE POLYNOMIALS OF D
C
      CALL SWEEP(NZD,RPZD,CPZD,AD,NAD)
      CALL SWEEP(NPD,RPPD,CPPD,BD,NBD)
      DO 402 I=1,NAD
      AD(I)=SD*AD(I)
  402 CONTINUE
C
      CALL PMPY(A,NA,AC,NAC,BD,NBD)
      CALL PMPY(B1,NB1,BC,NBC,BD,NBD)
      CALL PMPY(B2,NB2,AC,NAC,AD,NAD)
      CALL PADD(B,NB,B1,NB1,B2,NB2)
C
  320 CONTINUE
      TEMP=ABS(B(NB))
      DO 310 I=1,NB
      BB(I)=B(I)/TEMP
 310  CONTINUE
      BOA=BB(1)/A(1)
      DO 311 I=1,NA
```

```
      AA(I)=A(I)*BOA
311   CONTINUE
      RETURN
      END
```

```
C    **********************************************************************
C
C      WRITTEN BY WILLIAM R. GRIFFIN, MIT, DEPT OF AERO AND ASTRO
C      FEB 1970
C
C    **********************************************************************
C
C      SUBROUTINE SWEEP
C
C      PURPOSE
C         CALCULATES THE COEFFICIENTS OF A D.E. GIVEN THE ROOTS
C
C      USAGE
C         CALL SWEEP ( NR, RPR, CPR, D, ND )
C
C      DESCRIPTION OF PARAMETERS
C         NR - NUMBER OF ROOTS IN (COMPLEX PAIR = 1 ROOT)
C         RPR- REAL PART OF ROOT
C         CPR- COMPLEX PART OF ROOT (CONJUGATE NOT REQUIRED)
C         D  - COEFFICIENTS OUT D(1)=PROD(ROOTS)  D(ND-1)=-SUM(ROOTS)
C         D(ND)=1.0
C         ND - NUMBER OF COEFFICIENTS OUT
C
C      EXTERNAL ROUTINES REQUIRED
C         ABS
C
C    **********************************************************************
C
       SUBROUTINE SWEEP ( NR, RPR, CPR, D, ND )
       DIMENSION D(20),DD(20),RPR(20),CPR(20)
C
C--------INITIALIZE COEFFIENTS
C
       ND = 1
       DO 100 I=1,20
       D(I) = 0.
```

```
      DD(I) = 0.
  100 CONTINUE
      D(1) = 1.0
      IF(NR.EQ.0)GO TO 602
      DD(1) = 1.0
      J = 2
C
C--------CALCULATE POLYNOMIAL FROM ROOTS
C
      DO 600 I = 1, NR
C
C--------UPDATE POLYNOMIAL FOR REAL ROOTS
C
      IF (ABS(CPR(I)).GT.0.001) GO TO 350
      ND = ND + 1
      RS = -RPR(I)
C
C--------USE DUMMY ARRAY FOR TEMP STORAGE
C
      DO 200 KI = 1, J
      DD(KI + 1) = D(KI) + RS * D(KI + 1)
  200 CONTINUE
      DD(1) = D(1) * RS
C
C--------RETURN COEFFICIENTS BACK TO ORIGINAL ARRAY
C
      J = J + 1
      DO 300 KI = 1, J
      D (KI ) = DD (KI)
  300 CONTINUE
      GO TO 600
C
C--------UPDATE POLYNOMIAL FOR COMPLEX ROOT
C
  350 CONTINUE
      ND = ND + 2
```

```
      R2 = -2. * RPR(I)
      RS = RPR(I) * RPR(I) + CPR(I) * CPR(I)
C
C--------USE DUMMY ARRAY FOR TEMP STORAGE
C
      DO 400 KI = 1, J
 400  DD (KI + 2) = D(KI + 2) * RS + R2 * D(KI + 1) + D(KI)
      DD(2) = D(1) * R2 + D(2) * RS
      DD(1) = D(1) * RS
C
C--------RETURN COEFFICIENTS TO ORIGINAL ARRAY
C
      J=J+2
      DO 500 KI = 1, J
      D(KI) = DD(KI)
 500  CONTINUE
 600  CONTINUE
      I=0
 603  CONTINUE
      I=I+1
      TEMP=ABS(D(I))
      IF(TEMP.EQ.0.)GO TO 603
      DO 601 K=1,ND
 601  D(K) = D(K)/TEMP
 602  CONTINUE
      RETURN
      END
```

```
C ******************************************************************************
C
C       SUBROUTINE  INCON
C
C       PURPOSE
C
C       COMPUTES THE SYSTEM  INITIAL  CONDITIONS
C
C ******************************************************************************
C
C
      SUBROUTINE INCON(BCOF,ACOF,XO)
      DIMENSION ACOF(20),BCOF(20),XO(20)
      COMMON/CONST/N,M,K,KK,KKK,NN,NNN,NPO,NMO,NNPO,NNMO,NMON,NKK
      MI=M+1
      NM=N-M
      NMI=NM+1
      NMJ=NM+2
      XO(1)=-BCOF(1)/ACOF(1)
      DO 22 IK=2,NM
      XO(IK)=0.
   22 CONTINUE
      XO(NMI)=BCOF(MI)
      IF(M.LT.2) GO TO 61
      DO 25 IK=NMJ,N
      C=0.
      II=IK-1
      NI=N-IK+2
      DO 24 JK=NMI,II
      JN=JK+N-IK+1
      C=C-ACOF(JN)*XO(JK)
   24 CONTINUE
      XO(IK)=BCOF(NI)+C
   25 CONTINUE
   61 CONTINUE
      RETURN
      END
```

```
C    ***********************************************************************
C
C       HERMAN A. REDIESS,  MIT, DEPT. OF AERONAUTICS AND ASTRONAUTICS
C       MARCH 1968
C
C    ***********************************************************************
C
C       SUBROUTINE SNORM
C
C       PURPOSE
C          COMPUTES THE SUP-NORM OF AN NXN MATRIX OR AN NX1 VECTOR.
C
C       USAGE
C          CALL SNORM(A,B,N,MS)
C
C       DESCRIPTION OF PARAMETERS
C          A - NAME OF MATRIX OR VECTOR
C          B - NAME OF THE SUP-NORM OF A
C          N - DIMENSION OF A
C          MS - ONE DIGIT NUMBER FOR STORAGE MODE OF A
C                 0 - GENERAL
C                 1 - SYMMETRIC
C                 2 - DIAGONAL OR VECTOR
C
C       FUNCTIONS REQUIRED
C          ABS
C
C    ***********************************************************************
C
        SUBROUTINE SNORM(A,B,N,MS)
        DIMENSION A(1)
        DOUBLE PRECISION A
        B=0.
        IF(MS-1)10,11,12
     10 L=N*N
        GO TO 15
```

```
11  L=N*(N+1)/2
    GO TO 15
12  L=N
15  DO 20 I=1,L
    IF(DABS(A(I))-B)20,20,18
18  B=DABS(A(I))
20  CONTINUE
    RETURN
    END
```

```
C *************************************************************************
C
C       AN EXAMPLE OF SUBROUTINE ROOTS FOR A SIXTH ORDER SYSTEM
C
C
C       REMARK
C
C       NOTE THAT THE SIGN OF THE STATIC SENSITIVITY MUST BE REVERSED
C       FOR ANY BLOCK  WHOSE TRANSFER FUNCTION HAS AN ODD NUMBER OF
C       ZEROES IN THE RIGHT HALF COMPLEX PLANE
C
C *************************************************************************
C
C
      SUBROUTINE ROOTS (PAR,SI,MODE)
      DIMENSION RPPA(20),CPPA(20),RPZA(20),CPZA(20),PAR(10)
      DIMENSION RPZB(20),CPZB(20),RPPB(20),CPPB(20)
      DIMENSION RPPP(10),CPPP(10),RPZP(10),CPZP(10),SI(10)
      DIMENSION RPPC(10),CPPC(10),RPZC(10),CPZC(10)
      DIMENSION RPPD(10),CPPD(10),RPZD(10),CPZD(10)
      COMMON / ROOT / RPPA,CPPA,RPZA,CPZA,RPPB,CPPB,RPZB,CPZB,RPPC,CPPC,
     1RPZC,CPZC,RPPD,CPPD,RPZD,CPZD,RPPP,CPPP,RPZP,CPZP,SA,SB,SC,SD,SP,
     2NPA,NZA,NPB,NZB,NPC,NZC,NPD,NZD,NPP,NZP
      MODE=2
      NPA=4
      NZA=2
      NPB=0
      NZB=1
      RPPA(1)=-.01158*SI(1)
      CPPA(1)=2.315*SI(1)
      RPPA(2)=0.242
      CPPA(2)=0.
      RPPA(3)=-.294
      CPPA(3)=0.
      RPPA(4)=-.707*PAR(3)
      CPPA(4)=.707*PAR(3)
```

```
RPZA(1)=-.026+1.113*SI(1)
CPZA(1)=0.
RPZA(2)=-.026-1.113*SI(1)
CPZA(2)=0.
RPZB(1)=-PAR(2)
CPZB(1)=0.
SA=-1.03*PAR(1)
SB=1.
RETURN
END
```

APPENDIX D

References

1. Bode, H.W.; Network Analysis and Feedback Amplifier Design. Van Nostrand Co., N.Y., 1945.

2. Bradt, A.J.; "Sensitivity Functions in the Design of Optimal Controllers," IEEE Transactions on A.C., Vol.AC-13, Feb.1968.

3. Cadzow, J.A.; "Optimal Control of a System Subject to Parameter Variations," Proceedings of the Joint Automatic Control Conference, 1966.

4. Cassidy, J.F. and Roy, R.J.; "Sensitivity Design of Output Feedback Controllers," Proceedings of the Joint Automatic Control Conference, 1970.

5. Clark, R.N.; Introduction to Automatic Control Systems. J.Wiley Inc., 1962.

6. Cruz, J.B. and Perkins, W.R.; "A New Approach to the Sensitivity Problem in Multivariable Feedback System Design," IEEE Trans. on A.C., Vol. AC-9, July 1964.

7. D'Angelo, H., Moe, M.L. and Hendricks, T.C.; "Trajectory Sensitivity of an Optimal Control System," Proceedings of the Allerton Conference on Circuits and Systems Theory, 1966.

8. Dompe,R.J. and Dorf, C.D.; "The Design of Optimal Low Sensitivity Linear Control Systems," Proceedings of the 1967 Allerton Conference.

9. Dorato, P.; "On Sensitivity in Optimal Control System," IEEE Trans. on A.C., Vol. AC-8, July 1963.

10. Dougherty, H.J., Lee, I. and DeRusso, P.M.; "Synthesis of Optimal Feedback Control Systems Subject to Parameter Variations," Proceedings, Joint Automatic Control Conference, 1967.

11. Gibson, J.E., Rekasius, Z.V., McVey, E.S., Svidhar, R., Leedham, C.D.; "A Set of Standard Specifications for Linear Automatic Control Systems," Presented at the AIEE Winter General Meeting, N.Y., Jan. 1961.

12. Greensite, A.L.; "Analysis and Design of Space Vehicle Flight Control Systems," Vol. V, Sensitivity Theory, NASA Contractor Report, CR-824, 1967.

13. Grundy, P.A.; "The Matrix Equation AX+XB = -C and its Application to Control System Design," Master's thesis in the Dept. of Aeronautics and Astronautics, M.I.T., Jan.1971.

14. Higginbotham, R.E.; "Design of Sensitivity and State Variable Feedback Optimal Control Systems," Proceedings of the 1969 Allerton Conference.

15. Hendricks, T., and D'Angelo, H.; "An Optimal Fixed Control Structure Design with Minimal Sensitivity for a Large Elastic Booster," Proceedings of the 1967 Allerton Conference.

16. Horowitz, I.M.; Synthesis of Feedback Systems, Academic Press, 1963.

17. Horowitz, I.M.; "Fundamental Theory of Automatic Linear Feedback Control Systems," IRE Transactions on A.C., Vol. AC-4, Dec. 1959.

18. Horowitz, I.M.; "Plant Adaptive Systems Versus Ordinary Feedback Systems," IRE Trans. on A.C., Vol. AC-7, Jan. 1962.

19. Horowitz, I.M.; "Linear-Adaptive Flight Control Design for Re-entry Vehicles," IEEE Trans. on A.C., Vol. AC-9, Jan. 1964.

20. Horowitz, I.M.; "Optimum Linear Adaptive Design of Dominant Type Systems with Large Parameter Variations," IEEE Trans. on A.C., Vol. AC-14, June 1969.

21. Kahne, S.J.; "Low-Sensitivity Design of Optimal Linear Control Systems Subject to Parameter Variations," Air Force Flight Dynamics Laboratory Report, AFFDL-TR-66-209, April 1967.

22. Kreindler, E.; "Synthesis of Flight Control Systems Subject to Parameter Variations," Air Force Flight Dynamics Laboratory Report, AFFDL-TR-66-209, April 1967.

23. Lamont, G.B. and Kahne, S.J.; "Time Domain Sensitivity Comparisons," Proceedings of the 1967 Allerton Conference.

24. Mazer, W.H.; "Specifications of the Linear Feedback System Sensitivity Function," IRE Trans. on A.C., Vol. AC-5, June 1960.

25. Makers, D.T., Haake, H.B. and Briggs, P.; "Investigation of Advanced Flight Control Systems through Application of Linear Sensitivity Techniques," Air Force Flight Dynamics Laboratory Report, AFFDL-TR-66-142, Oct. 1966.

26. McRuer, D.T. and Stapleford, R.L.; "Sensitivity and Modal Response for Single-loop and Multi-loop Systems," Aeronautical Systems Division Report, ASD-TDR-62-812.

27. Newton, G.C., Gould, L.A. and Kaiser, J.F., Linear Feedback Controls, John Wiley Inc., 1957.

28. Pagurek, B.; "Sensitivity of the Performance of Optimal Linear Control Systems to Parameter Variations," Int. Journal of Automatic Control, Vol. 1, 1965.

29. Pagurek, B.; "Sensitivity of the Performance of Optimal Control Systems to Plant Parameter Variations," IEEE Trans. on A.C., vol. AC-10, April 1965.

30. Price, C.F. and Deyst, J.J.; "A Method for Obtaining Desired Sensitivity Characteristics with Optimal Controls," M.I.T. Experimental Astronomy Laboratory Report, RE-37, Jan. 1968.

31. Rediess, H.A.; "A New Model Performance Index for the Engineering Design of Control Systems," Doctoral Thesis in the Dept. of Aeronautics and Astronautics, M.I.T., Dec. 1968.

32. Rediess, H.A. and Whitaker, H.P.; "A New Model Performance Index for Engineering Design of Flight Control Systems," AIAA Journal of Aircraft, Vol. 7, Nov. 1970.

33. Rillings, J.H. and Roy, R.J.; "Analog Sensitivity Design of Saturn V Launch Vehicle," IEEE Trans. on A.C., Vol. AC-15, August 1970.

34. Rohrer, R.A. and Sobral, M.; "Sensitivity Considerations in Optimal System Design," IEEE Trans. on A.C., Vol.AC-10, Jan. 1965.

35. Rynaski, E.G., Whitbeck, R.F. and Wierwille, W.W.; "Optimal Control of a Flexible Launch Vehicle," Cornell Aeronautical Laboratory Report, CAL No. IH-2089-F-I.

36. Rynaski, E.G., Whitbeck, R.F. and Dolbin, B.H.; "Sensitivity Considerations in the Optimal Control of a Flexible Launch Vehicle," Cornell Aeronautical Laboratory Report, CAL No. BE-2311-F-1.

37. Sanutti, P. and Cruz, J.B.; "A Note on Trajectory Sensitivity of Optimal Control Systems," IEEE Trans. on A.C., Vol. AC-13, Feb. 1968.

38. Schmidt, G.T.; "A New Technique for Identification and Control of Systems with Unknown Parameters," Doctoral thesis in the Dept. of Aeronautics and Astronautics, M.I.T., Oct. 1970.

39. Stapleford, R.L. and McRuer, D.T.; "Sensitivity of Multiloop Flight Control System Roots to Open-Loop Parameter Variations," AIAA Journal, Vol. 4, Sept. 1966.

40. Swaim, R.L.; "Adaptive versus Non-Adaptive Control of Lifting Entry Vehicles," Paper no. 66-48, AIAA Aerospace Sciences Meeting, Jan. 1966.

41. Tomović, R.; Sensitivity Analysis of Dynamic Systems, McGraw-Hill, 1963.

42. Tomović, R.; "The Role of Sensitivity Analysis in Engineering Problems," Proceedings of the IFAC Conference on Sensitivity Methods in Control Theory, Dubrovnik, August 1964.

43. Tuel, W.G.; "Optimal Control of Unknown Systems," Doctoral thesis, Rensselaer Polytechnic Institute, June 1965.

44. Taylor, L.W. and Rediess, H.A.; "A Flight Control System Design Problem for the Design Challenge to the 1970 JACC," Presented at the Joint Automatic Control Conference, June 1970.

45. Thomas, H.J. and Lewis, J.B.; "An Optimal Controller for a Plant with Random Parameters," Proceedings of the 1969 Allerton Conference.

46. Ur, H.; "Root Locus Properties and Sensitivity Relations in Control Systems," IRE Trans. on A.C., Jan. 1960.

47. Whitaker, H.P. and Potter, J.E.; "Optimization of the Use of Automatic Flight Control Systems for Manned Aircraft," M.I.T. Instrumentation Laboratory Report R-558, August 1966.

48. Whitbeck, R.F.; "A Suboptimal Approach to the Control of Systems Containing Uncertain Parameters," AIAA Journal of Aircraft, Vol. 4, Nov. 1967.

BIOGRAPHICAL NOTE

Thorgeir Palsson was born in Reykjavík, Iceland on May 19, 1941. He was graduated from Menntaskólinn í Reykjavík in 1961. He attended the University of Iceland for one year before enrolling at Purdue University, Lafayette, Indiana. In 1963 he entered the Massachusetts Institute of Technology in the Department of Aeronautics and Astronautics where he studied in the honours program, receiving a Bachelor's degree in 1966, and a Master's degree in 1967 while holding a DuPont Fellowship.

In 1967 he joined the Measurement Systems Laboratory at M.I.T. where he has been involved in research in the fields of inertial guidance and flight control as a research assistant and staff member. He is the co-author of three papers and two reports in the area of guidance and control and has served as a consultant for the Northrop Corporation, Electronics Division.

Mr. Palsson is married to Anna Haraldsdottir of Reykjavík, Iceland. They have three daughters, Sigrun, Brynhildur and Elisabet.